International Journal of Artificial Intelligence in Education (2019) 29:459–483
https://doi.org/10.1007/s40593-019-00181-3

ARTICLE

Using Lexical Properties of Handwritten Equations to Estimate the Correctness of Students' Solutions to Engineering Problems

Thomas F. Stahovich[1] · Hanlung Lin[2,3] · Justin Gyllen[4]

Published online: 11 June 2019

Abstract

We present a technique that examines handwritten equations from a student's solution to an engineering problem and from this estimates the correctness of the work. More specifically, we demonstrate that lexical properties of the equations correlate with the grade a human grader would assign. We characterize these properties with a set of features that include the number of occurrences of various classes of symbols and binary and tripartite sequences of them. Support vector machine (SVM) regression models trained with these features achieved a correlation of $r = .433$ ($p < .001$) on a combined set of six exam problems. Prior work suggests that the number of long pauses in the writing that occur as a student solves a problem correlates with correctness. We found that combining this pause feature with our lexical features produced more accurate predictions than using either type of feature alone. SVM regression models trained using an optimized subset of three lexical features and the pause feature achieved an average correlation with grade across the six problems of $r = .503$ ($p < .001$). These techniques are an important step toward creating systems that can automatically assess handwritten coursework.

Keywords Educational data mining · Digital ink · Problem solving · Handwritten equations · Smartpen

✉ Thomas F. Stahovich
stahov@engr.ucr.edu

[1] Department of Mechanical Engineering, University of California, Riverside, CA, USA

[2] Department of Computer Science and Engineering, University of California, Riverside, CA, USA

[3] Amazon, Seattle, WA, USA

[4] Department of Mechanical Engineering, University of California, Riverside, CA, USA

Introduction

The ready availability of large amounts of data from educational software systems has enabled data mining techniques to be used to examine a wide range of education research questions (Romero and Ventura 2010). For example, log files from intelligent tutoring systems (e.g., Stevens et al. 2005) and learning management systems (e.g., Krüger et al. 2010) are common sources of data for mining. However, in many disciplines — particularly science, technology, engineering, and math (STEM) — learning involves a substantial amount of problem solving with paper and pencil, which is more challenging to mine than text-based work.

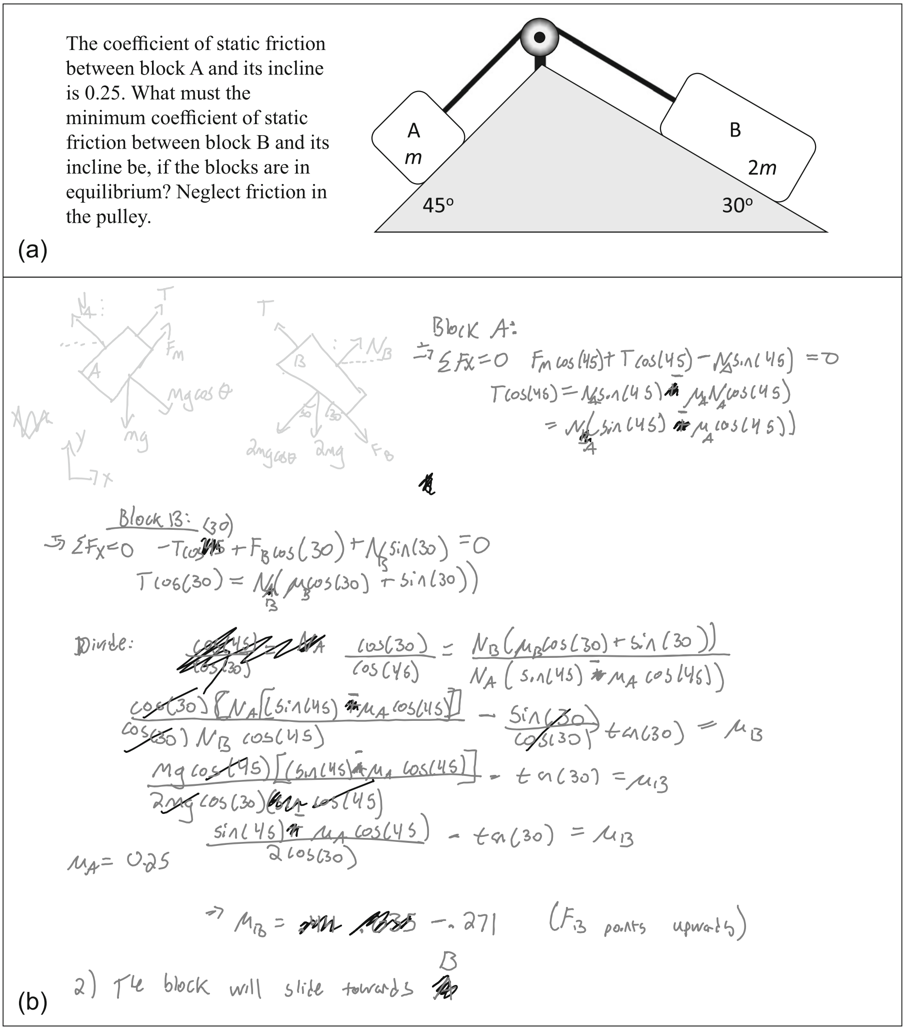
In previous work (Stahovich and Lin 2016), we developed techniques capable of extracting text information from handwritten solutions to engineering problems like the one in Fig. 1b. As an example of the utility of these methods, we used them to examine the relationship between the amount of writing in a student's handwritten solution to an exam problem and the correctness of the work. More specifically, we found that the total number of alphabetic characters (i.e., the 26 English characters), the number of units of measure (e.g., "kg" and "ft"), and the number of equation groups each correlated positively and significantly with the grade assigned by a human grader. An equation group is a string of characters belonging to a single equation and written on the same baseline (Fig. 2). This work also demonstrated that the number of long pauses between characters correlated positively and significantly with the grade.

This prior work primarily examined the relationship between the amount of writing and the correctness of a solution. Here, we examine the hypothesis that the types of content comprising a solution, and the sequences in which it is arranged, relate to the correctness. For example, skilled problem-solvers often solve problems by manipulating the equations in symbolic form, and avoid substituting numerical values into the variables until the final step. One advantage of this approach is that it facilitates the identification of errors. For example, while it is clear that "$F = m * v$" is an incorrect statement of Newton's second law (force relates to acceleration not velocity), it is not readily apparent if "F = 20.0 * 4.5" is a correct statement of this law. Likewise, manipulating symbolic variables reduces transcription errors that can occur when manipulating multi-digit real numbers. Thus, having a majority of non-numerical symbols rather than numbers may be indicative of correctness.

In short, the present work examines the hypothesis that lexical properties of a student's handwritten solution to a problem in a STEM course correlate with the correctness of the solution. We consider a number of lexical properties including the number of occurrences of various classes of symbols (e.g., letters, numbers, and mathematical symbols), the number of occurrences of various binary sequences of characters (e.g., a digit followed by a letter), and the number of tripartite sequences (e.g., a digit followed by a mathematical symbol followed by a letter). Likewise, we also consider the number of equation groups and the number of occurrences of units of measure from (Stahovich and Lin 2016). We refer to these as "lexical properties" to emphasize that we do not consider the semantics of the symbols. Said differently, we do not interpret the meaning of the written solution but rather consider only the quantities of various types of textual elements.

As the work in (Stahovich and Lin 2016) suggests that the number of long inter-character pauses that occur as a student solves a problem in a STEM course is related to the correctness of the solution, we include this feature in our models. Similar to the lexical features, this feature can be computed without interpreting the meaning of the solution.

For our present study, we used Livescribe smartpens to collect a dataset of handwritten solutions to exam problems from an undergraduate engineering course on statics. The smartpens have an integrated camera and are used with dot-patterned paper. They serve the same function as a traditional ink pen and also record the



The coefficient of static friction between block A and its incline is 0.25. What must the minimum coefficient of static friction between block B and its incline be, if the blocks are in equilibrium? Neglect friction in the pulley.

(a)
A
m
B
2m
45°
30°
(b)
N_A
T
F_m
A
mg cos θ
mg
y
x
T
N_B
B
30
30
2mg cos θ
2mg
F_B
Block A:
ΣFx=0 F_m cos(45) + T cos(45) − N_A sin(45) = 0
T cos(45) = N_A sin(45) − μ_A N_A cos(45)
= N_A(sin(45) − μ_A cos(45))
Block B:
ΣFx=0 −T cos(30) + F_B cos(30) + N_B sin(30) = 0
T cos(30) = N_B(μ_B cos(30) + sin(30))
Divide:
cos(30)/cos(45) = N_B(μ_B cos(30) + sin(30)) / N_A(sin(45) − μ_A cos(45))
cos(30)[N_A[(sin(45) − μ_A cos(45)]] / cos(30) N_B cos(45) − sin(30)/cos(30) tan(30) = μ_B
mg cos(45)[(sin(45) − μ_A cos(45)] / 2mg cos(30) cos(45) − tan(30) = μ_B
sin(45) − μ_A cos(45) / 2 cos(30) − tan(30) = μ_B
μ_A = 0.25
μ_B = −.271 (F_B points upwards)
2) The block will slide towards B

Fig. 1 A typical statics problem. **a** Problem statement. **b** A typical solution. Light gray = free body diagram, medium gray = equation, black = cross-out

work as time-stamped pen strokes, thus enabling both temporal and spatial analysis of the writing. Statics is the subdiscipline of engineering mechanics that examines the equilibrium of structures subject to forces. The solution to a statics problem typically includes free body diagrams and equilibrium equations. The former represent the forces acting on a system, while the latter are the application of Newton's Second Law. Figure 1a shows a typical problem from an undergraduate statics course and Fig. 1b shows the sort of solution a student might generate for that problem.

This work demonstrates that the lexical properties of handwritten solutions to a problem in an undergraduate engineering course are predictive of the correctness of the solution. This work could provide the basis for an automated system to provide students with feedback on their homework. In large undergraduate STEM courses, it is often impractical to manually grade students' homework. Our techniques provide an inexpensive and scalable means of estimating the correctness of this work. By examining the entire solution to a problem, our techniques complement traditional online homework systems that consider only the final answer (Demirci 2010). This sort of automated feedback would also be useful for online courses. While online courses provide an efficient means for delivering course content, there are currently no cost-effective methods for assessing handwritten work. Our techniques could provide the basis for creating such a method.

Related Work

Recent research has begun to examine the relationship between the amount of writing a student produces and academic achievement (Rawson et al. 2017; Van Arsdale and Stahovich 2012). For example, Rawson et al. (2017) examined students' writing on homework assignments in an introductory engineering course and found that the amount of writing, measured both in terms of the number of pen strokes and the length of ink written, correlated positively and significantly with course grade. Similarly, Van Arsdale and Stahovich (2012) found that the amount of effort on equations correlated positively and significantly with the correctness of the work. These studies examined the amount of writing, not the content, and found that it correlated positively with outcomes. In the present work, we build upon these results by examining how lexical properties of the content correlate with the correctness of a student's work.

Van Arsdale and Stahovich (2012) examined the relationship between the temporal and spatial organization of a student's handwritten solution to a statics problem and the correctness of the work. They computed 10 features describing the organization of the solution process and used them to construct stepwise regression models predicting the grade students achieved on the work. Our work is complementary in that we consider lexical properties of equations rather than the organization of the solution process.

Cheng and Rojas-Anaya (2008) examined pauses that occurred as students copied equations and found that the number of long pauses correlated negatively with competence. They defined a long pause as one longer than twice the median pause occurring while the student wrote his or her name. By contrast, Stahovich and Lin (2016) found that the number of long inter-character pauses during problem solving correlated positively with the correctness of the solution. The difference in the sign of the correlations is likely due to the nature of the tasks: one considers a copying task while the other considers a problem-solving task. We employ the pause measure from (Stahovich and Lin 2016) in the present work.

Research in educational data mining has seen a dramatic increase in the past few years (Romero and Ventura 2010). Much of the data used in this work is extracted from log files of intelligent tutoring systems (Stevens et al. 2005; Beal and Cohen 2008; Shanabrook et al. 2010; Mostow et al. 2011; Li et al. 2011; Trivedi et al. 2011) and learning management systems such as Moodle or Blackboard (Krüger et al. 2010; Romero et al. 2010). Our work differs from this in that we record and mine data from learning activities in natural environments, rather than online environments. The work of Oviatt et al. (2006) suggests that natural work environments are critical to student performance. In their examinations of computer interfaces for completing geometry problems, they found that "as the interfaces departed more from familiar work practice…, students would experience greater cognitive load such that performance would deteriorate in speed, attentional focus, meta-cognitive control, correctness of problem solutions, and memory."

While assessment is a critical element of effective instruction (Pellegrino et al. 2001; Bransford et al. 2000), it can be a burdensome task. Thus, educators have long sought to create methods for automating it. Gikandi et al. (2011) present a recent overview of online assessment tools. Multiple-choice exams are perhaps the most common automated offline tool. While such exams are inexpensive to grade, they generally capture the product of thinking rather than the process. Our techniques are complementary as they consider all of the work for a traditional handwritten problem, not just the final answer.

There have been some efforts to develop tools to facilitate manual grading of handwritten coursework (Schneider 2014; Singh et al. 2017), but there is relatively little work addressing automated grading. Recently, there has been some progress in developing systems for automatically grading handwritten essays (Srihari et al. 2007; Sharma and Jayagopi 2018). These systems first use optical handwriting recognition techniques to identify the text, and then apply automated essay scoring techniques to score the writing. As handwritten solutions to problems in STEM courses are dissimilar from essays, these techniques are not suitable for our task. One fundamental difference is that the text in an essay is written in a highly structured way (e.g., lines of text written from left to right and proceeding down the page), while the writing for a problem solution (e.g., Fig. 1b) is typically scattered around the page in a loosely structured fashion. Additionally, essays employ a known lexicon, whereas the combinations of symbols in a solution to a STEM problem are arbitrary. Researchers have developed techniques for interpreting handwritten equations (Smithies et al. 1999; LaViola and Zeleznik 2004; de Silva et al. 2007; LaViola 2007). These techniques are suitable for interpreting isolated equations and often require the user to draw in a structured manner or to use gestures to guide the interpretation. Thus, these techniques are unsuitable for our task, as the homework solutions we consider contain freeform writing.

Recently, Rawson and Stahovich (2013) and Rawson et al. (2017) examined the relationship between homework effort and course grade. Effort was represented by a set of features describing the amount of writing and the distribution of the writing activity over the assignment period. The features were used to construct regression models predicting course grade. These models demonstrated that the amount of writing correlated positively and significantly with course grade. Herold et al. (2013a) used a related approach that considered both the effort on individual problems and on the assignment as a whole. Herold et al. (2013b) represent homework activity as sequences of actions, including diagram drawing, equation writing, and taking breaks. They used differential data mining techniques to differentiate the activity sequences of students who achieved a high exam grade from those who achieved a low grade. All of these studies examined homework activity (effort) to predict future achievement in the course. By contrast, our work examines the lexical properties of equations written in solutions to exam problems to predict the correctness of the solutions.

Herold and Stahovich (2012) used smartpens as an assessment tool to examine how self-explanation affects the order in which students solve assigned homework problems. The study found that students who generated self-explanations of their work were more likely to finish each problem before starting the next compared to students who did not generate self-explanations.

More traditional educational data mining techniques have also been used to examine learning activities in statics courses. For example, work by Steif and Dollár (2009) examined usage patterns of a web-based statics tutoring system and found that learning gains increased with the number of tutorial elements completed. Similarly, work by Steif et al. (2010) examined whether students can be induced to talk about the bodies in a statics problem, and if doing so can increase a student's performance. They used tablet PCs to record the students' spoken explanations and their handwritten solutions, but the written work was left mostly unanalyzed.

Method

We used Livescribe smartpens to capture students' handwritten solutions to exam problems written on dot-patterned paper. The pens digitize pen strokes as they are written and store them as sequences of time-stamped Cartesian coordinates. We used techniques from Stahovich and Lin (2016) to process the pen stroke data into a form suitable for data mining. In the first step of processing, the equation pen strokes are separated from other content such as diagrams. Then the equation pen strokes are grouped, first into individual equations, and then into individual characters. Finally, after a character recognizer is used to recognize each individual character, a hidden Markov model is used to correct recognition errors.

Once the pen strokes have been recognized, we characterize a problem solution by computing features that characterize lexical properties of the equations. Some features describe the number of occurrences of various symbols and symbol combinations. One feature, for example, describes the number of occurrences of units of measure (e.g., "kg"), while another describes the number of occurrences of a letter following a mathematical operator. We also compute a feature counting the number of long inter-character pauses in the writing. We use support vector machine (SVM) regression models to relate these features to the correctness of the work. We take the grade assigned by a human grader to represent the correctness of a solution.

The next section describes the techniques we use to process the digital pen stroke data. This is followed by a description of our features and the dataset we used in this work.

Recognizing Equation Text

We use techniques from Stahovich and Lin (2016) to process the pen stroke data so that we can extract lexical features from it. Here we provide a brief summary of these techniques. Complete details can be found in (Stahovich and Lin 2016).

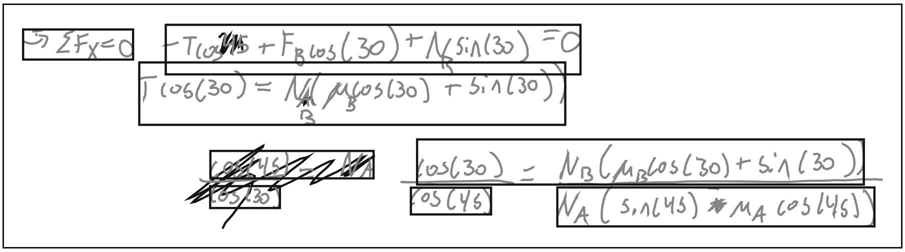
Handwritten solutions to engineering problems, like the one in Fig. 1b, contain a variety of content including diagrams, equations, and cross-outs. (Because the digital pens use ink which cannot be erased, students must cross out incorrect work.) The first step of processing is to identify which ink belongs to equations. This is accomplished with two filters. The first uses a set of heuristics to distinguish cross-outs from equations and diagrams. The second uses an AdaBoosted J48 decision tree, trained with a set of features describing the spatial and temporal properties of the pen strokes, to distinguish the equations from the diagrams.

Once the equation pen strokes have been identified, they are grouped into individual equation groups. As shown in Fig. 2, an equation group is a string of characters belonging to a single equation and written on the same baseline. One equation may comprise multiple equation groups. For example, if an equation wraps to a second baseline, there will be two equation groups, one for each baseline. Similarly, if a fraction is written with a horizontal fraction bar (vinculum), the numerator and denominator will likely be identified as separate equation groups.

We focus on equation groups, rather than complete equations, for the sake of simplicity. Identifying complete equations is a difficult problem for which no solutions currently exist. Consider, for example, the three equation groups in the lower right portion of Fig. 2. These three groups form a single equation:

$$\frac{\cos(30)}{\cos(45)} = \frac{N_B(\mu_B \cos(30) + \sin(30))}{N_A(\sin(45) - \mu_A \cos(45))} \tag{1}$$

However, identifying this would require complex semantic analysis of the writing. As the focus of our present study is to determine the relationship between lexical



→ ΣFx=0
-T(cos45 + F_B cos(30) + N_B sin(30) = 0
T cos(30) = N_A(μ_B cos(30) + sin(30))
cos(30) = N_B(μ_B cos(30) + sin(30))
cos(45)
N_A (sin(45) - μ_A cos(45))

Fig. 2 Rectangles indicate typical equation groups from Fig. 1b

properties of a student's handwritten solution — rather than semantic content — and the correctness, we avoid the complexity of the semantic analysis.

The equation grouper uses a classifier to determine if a pair of pen strokes belongs to the same equation group. The pairwise classifier is a J48 decision tree, implemented in WEKA (Hall et al. 2009) and trained using three features computed from the bounding boxes of the two pen strokes. Figure 3 shows the bounding boxes of two pen strokes and the four distances used to compute the features. The feature G_Y describes the vertical overlap of the bounding boxes. If y_A and y_B are the heights of the bounding boxes, and y_O is their vertical overlap, then $G_Y = \max(\frac{y_O}{y_A}, \frac{y_O}{y_B})$. G_Y is large if one of the characters lies mostly within the vertical extent of the other. A large value of G_Y suggests that the two pen strokes lie on the same baseline.

The feature G_D is related to the Manhattan distance. If x_D is the horizontal distance between the bounding boxes, G_D is defined as $x_D - y_O$. If the bounding boxes overlap horizontally, $x_D = 0$. G_D compares the horizontal spacing between two strokes to the vertical overlap between them. If the former is small compared to the latter, the strokes are near each other horizontally.

The feature G_{A2} is the ratio of the area of the intersection of the bounding boxes to the area of their union. However, before computing this ratio, the bounding boxes are expanded if they are too small. If the height of a bounding box is less than the median bounding box height, the box is expanded to that height. The width is adjusted analogously. Additionally, the width of each bounding box is then doubled to emphasize the horizontal arrangement of the strokes. The medians are computed separately for each problem solution. A large value of G_{A2} provides additional evidence that two pen strokes are near each other and are on the same baseline.

To group pen strokes into equation groups, the pairwise classifier is applied to every pair of strokes. A chaining process is then used to merge pairs of grouped strokes that share a common stroke. For example, if the pairwise classifier groups stroke A with B and B with C, the chaining process will combine A, B, and C into one group. Sometimes subscripts are not properly grouped with an equation. As a

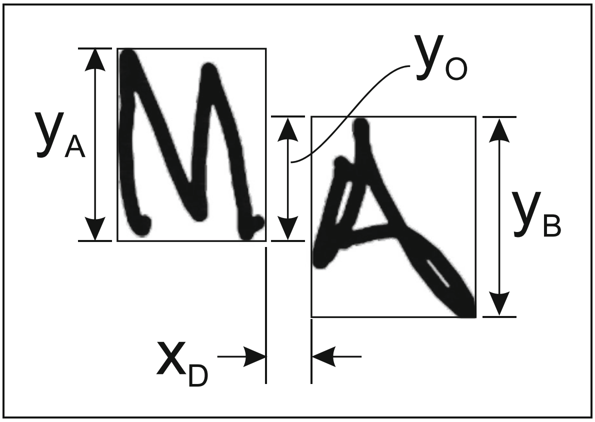


Fig. 3 Properties of bounding boxes used for grouping pen strokes into equation groups

remedy, small equation groups containing less than five strokes are merged with the nearest equation group if that group is nearby.

Once the pen strokes have been grouped into equations, it is necessary to group the strokes into individual characters so they can be recognized. For example, the letter “X” is typically drawn with two pen strokes. These two strokes must be grouped into a single multi-stroke character before the letter can be recognized.

Characters are grouped using a variation on the equation grouper employing only two features, G_A and G_X. G_A is similar to G_{A2}, but the widths of the bounding boxes are not doubled. G_X is similar to G_Y but considers horizontal overlap of the bounding boxes: $G_X = max(\frac{x_O}{x_A}, \frac{x_O}{x_B})$. Here x_A and x_B are the widths of the bounding boxes of the two strokes, and x_O is their horizontal overlap. As before, these features are used to train a J48 decision tree. This classifier is applied to all pairs of strokes in an equation group to determine which pairs form multi-stroke characters. Grouped pairs can chain together to form larger characters.

After the individual characters in a solution have been located, an image-based recognizer (Kara and Stahovich 2005) is used to recognize them. The recognizer uses a database of handwritten symbols to identify each character group. An approach based on a hidden Markov model (HMM) is used to correct recognition errors. Some errors are due to variations in writing style. Others result from ambiguity. For example, a lowercase “t” can be confused with a “+” and the number “1” can be confused with the letter “i”. The HMM uses local context to correct errors. For example, imagine that the recognizer identifies a sequence of characters as “s1n”. The HMM will examine the sequence and determine that “sin” is a more likely interpretation than “s1n”.

During error correction, the output of the image-based recognizer is considered to comprise the observations and the true identity of the characters are the hidden states. The Viterbi algorithm (Rabiner 1989) is used to compute the most likely sequence of hidden states to produce the observations. This sequence is then used as the interpretation of the equation.

Extracting Features from Equation Groups

Once the equations have been recognized, we compute 25 features from the text as summarized in Table 1. The first feature, F_E is the number of equation groups identified by the equation grouper.

Several features describe the number of occurrences of various classes of symbols. F_D is the number of individual digits in the solution (i.e., 0 – 9). F_L is the number of letters, including both the English alphabet and the Greek letters ‘θ’ and ‘ϕ’, which are often used to represent angles. We include only these two Greek letters (and ‘Σ’) because they occur far more frequently in our dataset than other Greek letters. F_M is the number of mathematical symbols including ‘+’, ‘-’, ‘*’, ‘/’, and ‘=’. The number of parentheses is excluded in the count of mathematical symbols. F_Σ is the number of occurrences of the symbol ‘Σ’, which is typically used in equation prototypes (see below). Finally, F_C is the total number of characters in the solution: $F_C = F_D + F_L + F_M + F_\Sigma + N_{()}$, where $N_{()}$ is the number of parentheses. (While we include the number of parentheses in the total count of characters, we found that

Table 1 Features for characterizing equations. D = digit, L = letter, M = mathematical symbol, "units" are units of measure, e.g., "kg" and "ft"

Features	Description
F_E	No. of equations
F_D	No. of digits
F_L	No. of letters
F_M	No. of mathematical symbols
F_Σ	No. of 'Σ' characters
F_C	No. of characters
$F_{D/L}$	Ratio of F_D to F_L
$F_{D/M}$	Ratio of F_D to F_M
$F_{L/M}$	Ratio of F_L to F_M
F_U	No. of units of measure
F_{DD}	No. of pattern DD
F_{DM}	No. of pattern DM
F_{DL}	No. of pattern DL
F_{LD}	No. of pattern LD
F_{LM}	No. of pattern LM
F_{LL}	No. of pattern LL
F_{MD}	No. of pattern MD
F_{MM}	No. of pattern MM
F_{ML}	No. of pattern ML
$F_{=D}$	No. of pattern $=D$
F_{DMD}	No. of pattern DMD
F_{DML}	No. of pattern DML
F_{LMD}	No. of pattern LMD
F_{LML}	No. of pattern LML
F_P	No. of long pauses

excluding this from the count of mathematical symbols resulted in slightly higher prediction accuracy.) Three features describe the relative number of occurrences of the three most common symbol classes: $F_{D/L} = F_D/F_L$, $F_{D/M} = F_D/F_M$, and $F_{L/M} = F_L/F_M$. Finally, F_U is the number of units of measure in the solution, including "kg", "g", "kN", "N", "m", "lb", "ft", and "in". To be identified as such, units must be immediately preceded by a digit such as "7 lb".

The next two categories of features are the number of occurrences of binary and tripartite sequences of digits (D), letters (L), and mathematical symbols (M). The features F_{ij} for $i, j \in \{D, L, M\}$ are the number of occurrences of binary sequences. For example, F_{DM} is the number of pairs of characters containing a digit followed by a mathematical symbol. The feature $F_{=D}$ considers the number of occurrences of the specific sequence in which an equal sign is followed by a digit, such as "= 4". Equal signs are important as they are one indication of the number of complete equations. The features F_{iMj} for $i, j \in \{D, L\}$ are the number of occurrences of tripartite sequences. For example, F_{DML} is the number of character sequences containing a digit, mathematical symbol, and letter, in that order.

Our set of lexical features is inspired by aspects of effective problem-solving approaches. For example, students in STEM courses are often encouraged to solve problems symbolically and then to plug in the numbers at the end. It is believed that manipulating symbols, rather than numbers, makes the concepts more evident to the student and reduces transcription errors. Likewise, students are encouraged to write units of measure (e.g., "kg" and "ft") for the various quantities when solving physics-based problems. Problem-solving errors often result in inconsistent or incorrect units. Thus, explicitly writing units can help students to identify errors. Similarly, when solving mechanics problems, students are encouraged to write equation prototypes such as "$\Sigma F_X = 0$", which is read as "the sum of the forces in the x-direction equals zero." Equation prototypes guide students in writing equilibrium equations. By representing the number of occurrences of the various classes of symbols, and the various combinations of them, our features model aspects of a student's problem-solving approach. Thus, we predict that these features will correlate with the correctness of the work.

The final feature, which is taken from (Stahovich and Lin 2016), characterizes the number of pauses between characters. The feature F_P is the number of inter-character pauses longer than the median inter-character pause.

Dataset

We used Livescribe smartpens to collect exam solutions from an undergraduate mechanical engineering course in statics taught at the University of California, Riverside. A total of 147 students enrolled in the course and 138 completed it. The course included two midterm exams and a final exam. Here we use data from the midterm exams. The data comprises a total of 1,069,918 pen strokes, of which 72% are equation strokes.

After we collected the midterm exam data, we manually partition it into individual problem solutions. To do this, we rendered each page of digital ink and interactively separated it by problem. In this way, we created a dataset containing 79 solutions for Midterm 1 Problem 1 (P1), 113 solutions for Midterm 1 Problem 2 (P2), 76 solutions for Midterm 1 Problem 3 (P3), 77 solutions for Midterm 2 Problem 1 (P4), 82 solutions for Midterm 2 Problem 2 (P5), and 48 solutions for Midterm 2 Problem 3 (P6).

The exam problems were graded by teaching assistants based on rubrics developed by the course instructor. These rubrics assigned credit for the correctness of individual problem-solving steps as well as the overall correctness of the solution. To verify the reliability of the grading, we randomly selected exams from 25 students and regraded the problems. As we did not have access to the rubric for problem P2, we did not regrade this problem. Also as not all students completed all exam problems, the random selection of 25 students resulted in only 22 solutions for 5 of the six exam questions. (There were 25 solutions for problem P1.) The new grades were highly consistent with the original ones. For problems P1, P3, P4, P5, and P6, the correlations between the original grades and the new grades were $r = .882$, $r = .896$,

r = .780, r = .932, and r = .926, respectively. These correlations are significant at $p < .001$.

Results

Table 2 shows the means and standard deviations for the 24 lexical features, the pause feature, and grade for each of the six exam problems. By some measures, students produced the least amount of equations for problem P2, and the most for problem P4. For example, the average number of equation groups (F_E) for problem P2 is 23.7 and for problem P4 it is 32.1. Likewise, the average number of characters written (F_C) for problem P2 is 227.8 and for problem P4 it is 329.1. Interestingly, problem P2 had the lowest average grade of 11.2, while problem P4 had the highest average grade of 16.4. (All problems have a maximum possible grade of 20.) The average number of long pauses (F_P) ranged from 45.1 for problem P5 to 69.8 for problem P4. Once again, problem P4 had the largest number of long pauses out of all six problems.

To examine our hypothesis that lexical properties of handwritten solutions correlate with the correctness of the work, we computed Pearson correlations between each of the 24 lexical features and grade for all six problems, both separately and combined. The results are listed in the first 24 rows of Table 3. For the six problems combined (column P:All), all of the correlations are positive and significant. In fact, for 22 of the lexical features, $p < .001$. (Note that all p values are computed with two tails and the number of degrees of freedom equal to the number of data points minus two.)

For four of the individual problems, the correlations with grade are significant for most of the lexical features. More specifically, for problem P1, all lexical features except F_Σ, $F_{D/L}$, $F_{D/M}$, and $F_{L/M}$ correlate positively and significantly with grade. For problem P2, all except F_Σ, $F_{D/L}$, and $F_{L/M}$ correlate positively and significantly with grade. For problem P3, all except $F_{L/M}$ correlate positively and significantly grade. For problem P6, all except F_Σ, $F_{D/M}$, $F_{L/M}$, and F_{LML} correlate positively and significantly with grade. $F_{L/M}$ correlates significantly, but the correlation is negative.

For problem P4, only three lexical features correlate significantly with grade: F_E correlates negatively and $F_{D/M}$ and $F_{L/M}$ correlate positively. For problem P5 only one lexical feature correlates significantly with grade: F_{DD} correlates positively.

Table 3 also includes Pearson correlations between the number of long pauses (F_P) and grade for all six problems, both separately and combined. The results are listed in the last row of Table 3. For the six problems combined, the correlation is positive and significant (r = .461, $p < .001$). Furthermore, F_P correlates positively and significantly with grade for all individual problems except problem P4. The average correlation coefficient across all six individual problems is r = .405.

As a measure of the collective power of the features for predicting grade, we used them to construct SVM regression models. We began by considering a problem-dependent training approach in which the model for each individual problem was trained and tested using data from only that problem. We constructed the models using WEKA's SVM regression method (SMOreg) with default parameter values

Table 2 Means and standard deviations of the lexical and pause features and grade for the six problems

	P1 (n=79)		P2 (n=113)		P3 (n=76)		P4 (n=77)		P5 (n=82)		P6 (n=48)	
Feature	M	SD	M	SD	M	SD	M	SD	M	SD	M	SD
F_E	30.5	13.6	23.7	12.5	27.0	12.9	32.1	13.5	26.8	11.7	26.1	10.9
F_D	117.4	56.3	71.8	41.5	96.2	51.3	106.7	36.1	76.8	38.0	90.0	46.9
F_L	106.5	44.2	78.0	42.7	98.7	45.7	117.9	39.2	78.8	35.4	85.6	43.7
F_M	77.5	34.0	60.5	30.9	73.6	35.9	80.3	30.3	59.3	25.2	60.2	29.1
F_Σ	2.9	1.6	3.1	1.9	4.7	2.3	4.9	2.3	3.5	2.2	2.7	1.8
F_C	328.7	136.4	227.8	117.3	293.9	134.6	329.1	105.4	233.1	101.6	261.0	126.2
$F_{D/L}$	1.1	0.4	1.0	0.3	1.0	0.3	0.9	0.2	1.0	0.3	1.1	0.3
$F_{D/M}$	1.5	0.4	1.2	0.4	1.3	0.4	1.4	0.3	1.3	0.4	1.5	0.4
$F_{L/M}$	1.4	0.3	1.3	0.4	1.4	0.3	1.5	0.3	1.3	0.3	1.5	0.3
F_U	5.4	3.0	3.0	2.4	5.3	4.1	7.0	3.5	2.6	1.9	4.4	3.5
F_{DD}	59.2	31.7	32.1	22.0	46.9	29.9	47.4	18.0	36.4	20.8	43.2	25.3
F_{DM}	28.8	13.9	19.1	11.1	23.9	12.6	26.7	10.8	19.1	10.0	21.4	11.6
F_{DL}	23.2	12.5	15.8	9.4	19.6	11.6	25.6	9.7	14.6	8.1	19.8	11.9
F_{LD}	16.7	9.1	8.5	6.2	12.7	7.5	18.3	7.3	9.4	6.3	12.5	7.9
F_{LM}	35.9	15.5	28.7	15.1	36.6	17.7	36.0	14.3	27.2	12.0	29.6	15.7
F_{LL}	41.9	18.5	31.4	19.9	39.7	20.3	51.7	18.8	31.5	16.1	33.3	18.0
F_{MD}	34.1	15.8	25.5	14.6	31.7	15.8	34.8	13.3	24.4	12.2	28.6	15.4
F_{MM}	20.7	10.7	15.0	10.1	20.5	12.3	20.6	10.2	13.0	7.9	16.8	9.7
F_{ML}	31.3	13.8	22.9	12.2	30.3	14.6	30.1	11.7	24.1	10.4	23.4	12.9
$F_{=D}$	18.6	9.7	14.1	8.7	17.2	9.4	19.8	8.2	15.1	7.9	16.6	10.1
F_{DMD}	11.1	6.3	7.4	5.4	8.4	5.9	9.2	4.9	6.7	4.2	8.0	5.0
F_{DML}	7.9	4.6	4.7	3.5	6.7	4.3	7.2	3.9	5.5	3.5	5.0	2.9
F_{LMD}	12.6	6.4	9.7	6.1	12.5	6.5	13.0	6.2	10.5	5.6	10.8	6.6
F_{LML}	10.4	5.5	8.5	5.6	10.0	6.5	9.6	5.4	7.7	4.2	8.0	5.9
F_P	69.3	27.5	47.4	24.5	63.7	27.3	69.8	22.1	45.1	18.5	53.8	25.1
Grade	13.2	4.1	11.2	3.5	11.5	4.4	16.4	3.6	12.1	5.1	12.0	4.5

All problems have a maximum possible grade of 20

Table 3 Correlations between the lexical and pause features and grade for the six problems separately and combined (P:All)

Feature	P1	P2	P3	P4	P5	P6	P:All
F_E	.365‡	.269†	.480‡	−.323†	−.016	.516‡	.256‡
F_D	.515‡	.498‡	.651‡	.011	.193	.487‡	.443‡
F_L	.455‡	.450‡	.633‡	−.045	.165	.385†	.418‡
F_M	.449‡	.374‡	.667‡	−.205	.144	.516‡	.372‡
F_Σ	.084	.049	.471‡	−.185	.156	.186	.190‡
F_C	.519‡	.472‡	.693‡	−.071	.196	.507‡	.446‡
$F_{D/L}$	.118	.172	.239*	.092	.149	.337*	.147†
$F_{D/M}$	.150	.282†	.237*	.340†	.212	.170	.250‡
$F_{L/M}$	−.046	.079	−.063	.267*	.052	−.296*	.096*
F_U	.359†	.387‡	.573‡	.154	.154	.347*	.414‡
F_{DD}	.491‡	.491‡	.577‡	.094	.229*	.461‡	.417‡
F_{DM}	.488‡	.432‡	.658‡	−.114	.146	.481‡	.396‡
F_{DL}	.468‡	.407‡	.591‡	.125	.163	.427†	.433‡
F_{LD}	.351†	.502‡	.624‡	.098	.120	.307*	.425‡
F_{LM}	.478‡	.368‡	.603‡	−.104	.185	.422†	.353‡
F_{LL}	.411‡	.448‡	.597‡	.026	.151	.315*	.414‡
F_{MD}	.541‡	.395‡	.648‡	−.087	.171	.501‡	.405‡
F_{MM}	.383‡	.255†	.569‡	−.096	.138	.626‡	.328‡
F_{ML}	.385‡	.390‡	.602‡	−.184	.172	.398†	.337‡
$F_{=D}$	.507‡	.383‡	.628‡	−.088	.121	.449†	.380‡
F_{DMD}	.511‡	.451‡	.553‡	−.166	.195	.407†	.356‡
F_{DML}	.365‡	.372‡	.565‡	−.057	.052	.407†	.321‡
F_{LMD}	.442‡	.365‡	.526‡	−.008	.179	.387†	.341‡
F_{LML}	.361†	.357‡	.402‡	−.117	.150	.167	.242‡
F_P	.481‡	.510‡	.746‡	−.087	.238*	.542‡	.461‡

$^{*}p < .05$, $^{\dagger}p < .01$, $^{\ddagger}p < .001$

(Hall et al. 2009). This method normalizes the data and uses a polynomial kernel with an exponent of 1.0. We trained the models for all problems, both separately and combined, using 10-fold cross-validation. For this training approach, the dataset is split into 10 equal-size, disjoint subsets. During each of the 10 folds, a model is trained using nine of the subsets, and that model is then used to make predictions for the remaining subset. At the completion of this process, there is a predicted grade for each data point. We characterize the performance of the models in terms of the Pearson correlation (r) between the predicted and actual grades, the root-mean-square error (RMSE) of the predictions, and the mean-absolute error (MAE). The results are listed in Table 4. The rows labeled "P:All" are the results for the six problems combined, while the rows labeled "Ave" are the average performance measures for the six individual problems.

Table 4 Correlations between actual and predicted grades for SVM regression models trained using problem-dependent training

Problem	Statistic	All	F_C	Lexical	Single	Double	Triple	F_P
	r	.404‡	.497‡	.414‡	.378‡	.478‡	.482‡	.447‡
P1	RMSE	4.00	3.58	3.95	4.03	3.68	3.65	3.75
	MAE	3.33	2.98	3.30	3.38	3.02	3.03	3.10
	r	.342‡	.464‡	.341‡	.329‡	.451‡	.434‡	.492‡
P2	RMSE	3.57	3.14	3.57	3.59	3.18	3.20	3.11
	MAE	2.69	2.51	2.67	2.61	2.52	2.58	2.47
	r	.565‡	.687‡	.552‡	.626‡	.629‡	.574‡	.728‡
P3	RMSE	3.81	3.24	3.90	3.55	3.56	3.73	3.03
	MAE	3.03	2.51	3.06	2.84	2.71	2.95	2.45
	r	.202	−.131	.155	.319†	.145	.025	−.110
P4	RMSE	4.00	3.81	4.11	3.57	3.76	3.89	3.78
	MAE	3.34	3.04	3.43	2.87	2.99	2.93	2.97
	r	−.030	.152	−.009	.081	−.009	.199	.157
P5	RMSE	6.05	5.19	5.93	5.75	5.71	5.18	5.24
	MAE	4.75	4.20	4.75	4.58	4.47	4.09	4.17
	r	.455†	.403†	.470‡	.446†	.482‡	.320*	.431†
P6	RMSE	4.38	4.10	4.31	4.08	4.03	4.25	4.12
	MAE	3.47	3.44	3.35	3.25	3.24	3.61	3.25
	r	.456‡	.435‡	.433‡	.453‡	.440‡	.381‡	.451‡
P:All	RMSE	4.10	4.11	4.16	4.09	4.11	4.22	4.08
	MAE	3.28	3.32	3.33	3.29	3.34	3.44	3.29
	r	.323	.345	.321	.363	.363	.339	.357
Ave	RMSE	4.30	3.84	4.29	4.10	3.99	3.98	3.84
	MAE	3.43	3.11	3.43	3.26	3.16	3.20	3.07

P:All = all problems combined into one dataset. All = all features, F_C = character count, Lexical = all lexical features, Single = single item counts, Double = binary pattern counts, Triple = tripartite pattern counts, F_P = long pause count. $^{*}p < .05$, $^{\dagger}p < .01$, $^{\ddagger}p < .001$. Ave = averages for the six problems. Significance not reported for average correlations

The "All" column in the Table 4 lists the correlations achieved using all of the features: the 24 lexical features and the pause feature. For all problems combined, the correlation with grade is .456 ($p < .001$), the RMSE is 4.10, and the MAE is 3.28. (When interpreting RMSE and MAE, note that grades range from 0.0 to 20.0). Additionally, the models correlate positively and significantly with grade for four of the six individual problems: P1 (r = .404, $p < .001$), P2 (r = .342, $p < .001$), P3 (r = .565, $p < .001$), and P6 (r = .455, $p < .001$). The average correlation coefficient across all six individual problems is r = .323, the average RMSE is 4.30, and the average MAE is 3.43.

To examine the relative predictive power of the various types of features, we trained SVM regression models using subsets of them. We considered five subsets: (A) F_C which comprises the total number of characters, (B) *Lexical* features which comprise the complete set of 24 lexical feature, (C) *Single* features which comprise single item counts $\{F_E, F_D, F_L, F_M, F_\Sigma, F_C, F_{D/L}, F_{D/M}, F_{L/M}, F_U\}$, (D) *Double* features which comprise binary pattern counts $\{F_{DD}, F_{DM}, F_{DL}, F_{LD}, F_{LM}, F_{LL}, F_{MD}, F_{MM}, F_{ML}, F_{=D}\}$, (E) *Triple* features which comprise tripartite pattern counts $\{F_{DMD}, F_{DML}, F_{LMD}, F_{LML}\}$, and (F) F_P which comprises the number of long pauses. These results are listed in Table 4. All six subsets produce models that correlate positively and significantly ($p < .001$) with grade for the six problems combined. For F_C r = .435, for the *Lexical* features r = .433, for the *Single* features r = .453, for the *Double* features r = .440, for the *Triple* features r = .381, and for F_P r = .451.

All six feature subsets produce models that correlate positively and significantly with grade for individual problems P1, P2, P3, and P6. Additionally, the models trained with the *Single* feature subset also correlate positively and significantly with grade for problem P4. For F_C, the average correlation with grade across all six problems is r = .345, for the *Lexical* features it is r = .321, for the *Single* features it is r = .363, for the *Double* features it is also r = .363, for the *Triple* features it is r = .339, and for F_P it is r = .357.

Note that the correlations for F_C for the six individual problems listed in Table 4 are smaller than the correlations for F_C listed in Table 3. The former are correlations between predicted grades and actual grades using a cross-validation approach in which the training and testing data are disjoint so as to reduce over-fitting. By contrast, the latter are direct correlations between F_C and grade.

The results in Table 4 characterize the performance of the models for problem-dependent training. Here, to explore the robustness of the models, we evaluate their performance using a problem-independent training approach. More specifically, when testing a model on data from a particular problem, we train the model on data from the other five problems. This training approach corresponds to a usage scenario in which models trained from previous problems are used to estimate grades on a new problem. The performance of these models is described in Table 5.

When using the problem-independent approach, the models trained using all features as well as the *Lexical*, *Single*, and *Double* feature subsets correlate positively and significantly with grade for problems P1, P2, P3, and P5. The models trained using the F_C, *Triple*, and F_P feature subsets correlate significantly with grade for all six problems: for problem P4 the correlations are negative and for the other five problems they are positive. For all features, the average correlation with grade across all six problems is r = .305, for F_C it is r = .285, for *Lexical* features it is r = .302, for *Single* features it is r = .336, for *Double* features it is r = .325, for *Triple* features it is r = .240, and for F_P it is r = .296. As is expected, the correlations are smaller, and the RMSE and MAE are larger for the problem-independent training than for the problem-dependent training.

Table 5 Correlations between actual and predicted grades for SVM regression models trained using problem-independent training

Problem	Statistic	All	F_C	Lexical	Single	Double	Triple	F_P
	r	.380‡	.431‡	.381‡	.422‡	.409‡	.358†	.457‡
P1	RMSE	4.36	4.18	4.36	4.32	4.25	4.35	4.16
	MAE	3.52	3.38	3.51	3.49	3.43	3.53	3.36
	r	.398‡	.415‡	.379‡	.387‡	.428‡	.357‡	.425‡
P2	RMSE	4.36	4.37	4.41	4.39	4.30	4.53	4.31
	MAE	3.61	3.61	3.66	3.64	3.57	3.76	3.56
	r	.459‡	.410‡	.449‡	.437‡	.432‡	.342†	.426‡
P3	RMSE	4.85	4.70	4.74	4.72	4.66	4.94	4.92
	MAE	3.89	3.75	3.81	3.78	3.76	3.94	3.94
	r	.007	$-.477^{\ddagger}$	.008	.118	$-.031$	$-.370^{\ddagger}$	$-.491^{\ddagger}$
P4	RMSE	6.45	7.36	6.45	6.20	6.57	7.59	7.47
	MAE	5.24	6.08	5.27	5.07	5.38	6.25	6.16
	r	.371‡	.494‡	.385‡	.441‡	.434‡	.383‡	.505‡
P5	RMSE	4.62	3.92	4.57	4.07	4.69	4.41	4.14
	MAE	3.72	3.18	3.67	3.32	3.74	3.53	3.36
	r	.216	.438†	.206	.209	.278	.367*	.451†
P6	RMSE	5.66	4.18	5.73	5.01	4.80	4.32	4.20
	MAE	4.47	3.33	4.51	3.96	3.87	3.51	3.35
	r	.305	.285	.302	.336	.325	.240	.296
Ave	RMSE	5.05	4.79	5.04	4.79	4.88	5.02	4.87
	MAE	4.07	3.89	4.07	3.88	3.96	4.09	3.96

All = all features, F_C = character count, Lexical = all lexical features, Single = single item counts, Double = binary pattern counts, Triple = tripartite pattern counts, F_P = long pause count. $^{*}p < .05$, $^{\dagger}p < .01$, $^{\ddagger}p < .001$. Ave = averages for the six problems. Significance not reported for average correlations

Using too many features in a model often results in over-fitting of the data. Here we examine optimal subsets of the features. We exhaustively enumerated and evaluated all possible models employing three lexical features and the pause feature. We trained these models in a problem-dependent fashion using 10-fold cross validation. Table 6 lists the optimal combination of features for each problem and the corresponding correlation coefficient, RMSE, and MAE. Table 7 contains the coefficients for the optimal regression models. Note that the models are computed using normalized feature values. For all six problems, the correlations are positive and significant. The average correlation across all six problems is $r = .503$, the average RMSE is 3.60, and the average MAE is 2.89. Figure 4 shows plots of residuals vs. predicted grades for the optimal models.

Table 6 Performance of optimal models constructed using three lexical features and the pause feature

Problem	Optimal features	r	RMSE	MAE
P1	$F_{D/L}$ F_{DD} F_{MD} F_P	.529‡	3.50	2.90
P2	F_{LD} F_{MD} F_{DMD} F_P	.534‡	2.99	2.38
P3	F_M F_U F_{DMD} F_P	.737‡	3.00	2.33
P4	F_E $F_{L/M}$ F_{DD} F_P	.396‡	3.33	2.66
P5	F_E $F_{D/L}$ F_{LML} F_P	.236*	5.07	4.15
P6	$F_{D/M}$ $F_{L/M}$ F_{MM} F_P	.584‡	3.68	2.90
P:All	$F_{D/M}$ F_U F_{DML} F_P	.484‡	4.00	3.23
Ave		.503	3.60	2.89

SVM regression models trained using 10-fold cross-validation. *$p < .05$, $^†p < .01$, $^‡p < .001$. P:All = all problems combined. Ave = average for the six problems. Significance not reported for the average correlation

Discussion and Future Work

Our results support our prediction that the lexical properties of a student's handwritten solution to a problem in a STEM course correlate with the correctness of the solution. We found that all of the lexical features correlate positively and significantly ($p < .001$) with grade for the six problems combined. Furthermore, for four of the six individual problems (P1, P2, P3, and P6), nearly all of the lexical features correlate positively and significantly with grade.

SVM regression models trained in a problem-dependent fashion (i.e., with training and testing data comprising disjoint subsets of data from the same problem) demonstrated that the lexical features, in combination, are predictive of the correctness of a handwritten solution (Table 4). For example, models trained with the complete set of lexical features, as well as those trained with four different subsets of the lexical

Table 7 Optimal SVM regression models for computing grade

Problem	Grade=				
P1	+0.2497	+0.4199F_{MD}	+0.2319F_{DD}	−0.1865$F_{D/L}$	+0.1560F_P
P2	+0.4121	+0.5069F_{DMD}	+0.3603F_{LD}	+0.2238F_P	−0.1947F_{MD}
P3	+0.0537	+0.7484F_P	+0.2145F_U	+0.1677F_M	−0.1304F_{DMD}
P4	+0.7394	−0.6847F_E	+0.5548F_{DD}	+0.2843$F_{L/M}$	−0.1943F_P
P5	+0.2521	−0.5216F_E	+0.4262F_P	+0.4220F_{LML}	+0.3925$F_{D/L}$
P6	+0.1832	+0.4922F_{MM}	+0.2958$F_{D/M}$	−0.2618$F_{L/M}$	+0.1853F_P
P:All	+0.2319	+0.5427F_P	+0.3393$F_{D/M}$	+0.1494F_U	−0.0168F_{DML}

The models use *normalized* feature values and are computed using the entire set of examples without cross-validation

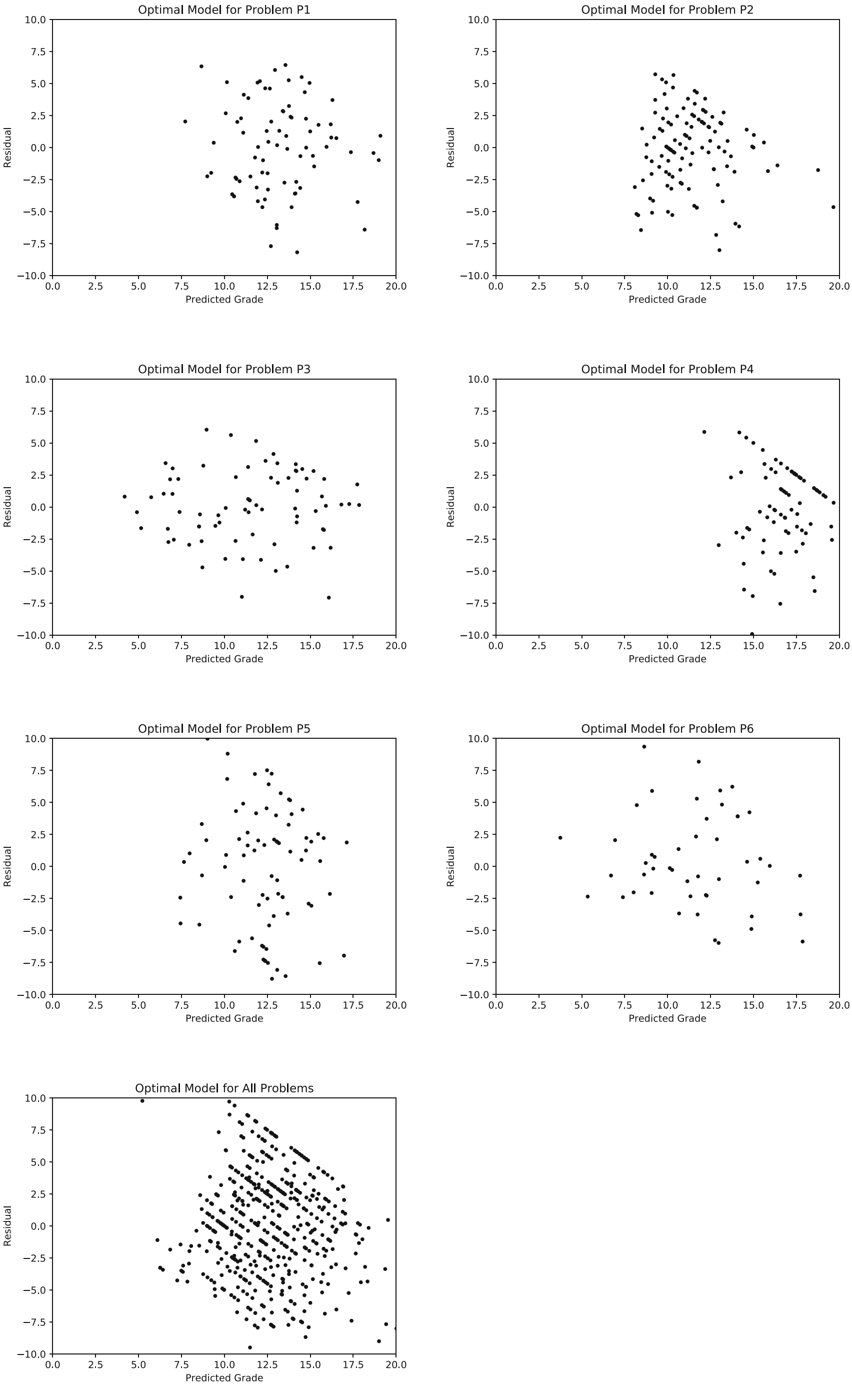


Fig. 4 Residual vs. predicted grade for the optimal SVM models trained using three lexical features and the pause feature. (See Table 6)

features (the F_C, *Single*, *Double*, and *Triple* subsets), all correlate positively and significantly ($p < .001$) with grade for the six problems combined.

Prior work in (Stahovich and Lin 2016) demonstrated that the number of long pauses in a student's handwritten solution to a problem in a STEM course correlates with correctness. Our results demonstrate that our lexical features provide information beyond that provided by the pause feature. In fact, combining the lexical and pause features produced the best performance. For example, the models in Table 6, which each include three lexical features and the pause feature, performed better than the models in Table 4, which comprised only subsets of the lexical features or only the pause feature.

Because of their nature, the various lexical features are correlated with each other. For example, as the total number of characters increases, the number of digits, letters, and mathematical symbols each typically increase as well. Indeed, SVM regression models trained in a problem-dependent fashion using only the number of characters (F_C) did correlate positively and significantly with grade for four of the six individual problems (P1, P2, P3, and P6) and for all problems combined. However, the other features do provide additional information as is evident from the optimal models in Table 6. The feature F_C was not selected in any of the optimal feature models. Thus, beyond the number of characters, the number of occurrences of the various classes of symbols and binary and tripartite sequences of them are important features for assessing the correctness of a problem.

The correlations for problem-independent training (Table 5) are somewhat smaller than the correlations for problem-dependent training (Table 4). Nevertheless, the fact that problem-independent training produces significant correlations suggests that the methods may be useful in scenarios in which models are trained on existing problems are then used to estimate grades on new problems. However, there are clearly limits to the generality of the models, and exploring this is left to future work.

For both problem-dependent and problem-independent training, models trained using the full set of lexical features failed to produce significant correlations for problem P4. (However, the optimal model did produce a significant correlation with grade.) We suspect that this may be related to the nature of this particular exam question. This problem had the highest average grade out of the six problems. On average, students received 16.4 out of a possible 20 points. We suspect that the weak correlations for this problem are a result of a large number of students performing particularly well so that the distribution of grades was highly skewed. In total, 32% of students received a perfect grade of 20.

For problem P5, problem-dependent training using the full set of lexical features failed to produce a significant correlation. However, problem-independent training with the full set of lexical features did produce a significant correlation, as did the optimal model. This may suggest a problem with over-fitting.

Figure 4 shows plots of the residuals vs. predicted grades for the optimal models from Table 6. For the most part, the residuals are unbiased. The residuals for problem P2 are somewhat heteroscedastic, but this may be a result of the dearth of examples with high grades. The diagonal band in the upper right of the residual plot for problem P4 is a result of the high proportion of students who received perfect grades. Each of these points represents a student who received a perfect grade, and thus the predicted values and residuals are linearly related. This same band appears in the plot for all problems combined.

We believe that our features are predictive of correctness because they characterize aspects of effective problem solving. For example, by characterizing the relative frequency of non-numerical symbols vs. numbers, our features may detect when a student works symbolically and delays the use of numbers until the last step, which is an effective approach to problem solving. Nevertheless, we avoid attempting to interpret the coefficients of the SVM regression models in Table 7 as their meaning is not clear.

Instead, our models are best evaluated in terms of their accuracy at making predictions. Our regression results characterize prediction accuracy as the training and testing data for the models was distinct: the results in Table 4 describe the prediction performance for problem-dependent training using cross-validation, while the results in Table 5 describe performance for problem-independent training in which the training and testing data are form different problems. Said differently, our results characterize performance at extrapolation rather than interpolation.

The best prediction accuracy was achieved by the optimal models described in Table 6, which were also trained in a problem-dependent fashion using cross-validation. For the six individual problems, the models achieved an average correlation coefficient $r = .503$, an average RMSE of 3.60 and an average MAE of 2.89. On average, these models explained 25% of the variance in the grades ($r^2 = .253$) and, thus are capable of making useful predictions of grades. However, as the RMSE is 3.60 on a grading scale of zero to 20, the predictions are not yet sufficiently accurate for automated grading. The models are best used for providing automated feedback to students. For example, in cases where it is impractical to grade student homework (which unfortunately occurs all too often in large STEM courses), the models could be used to identify students who have poor predicted grades on multiple problems. Those students could then be given additional support with the material. For this application, erroneously low predicted grades would cause no harm.

While an average correlation coefficient of $r = .503$ is still insufficient for automated grading, these results are none the less surprising. The models do not attempt to interpret a student's equations or the final answer. In fact, the models do not even consider if a final answer exists. The predictions are based solely on lexical characteristics of the writing and the number of long pauses. We believe that there may be other lexical properties of handwritten equations, not considered by our feature set, that also correlate with correctness. Identifying these could improve prediction accuracy, but that is future work. Likewise, our work is complementary to that of Van Arsdale and Stahovich (2012) who used features characterizing the temporal and spatial organization of a student's handwritten solution to predict correctness. We expect that combining our features with theirs will produce even more accurate predictions.

It is useful to contrast our task with another automated grading task: automated essay scoring (AES). AES techniques are quite mature. For example, Attali (2015) reported correlations between machine generated scores and human generated scores as high as $r = .79$. However, this task is considerably different from ours. AES systems work with machine interpretable text, while we work from handwritten pen strokes. Recently, Sharma and Jayagopi (2018) developed a method for automated grading of handwritten essays. They formulated the problem as the task of classifying an essay with one of five possible integer scores in the range from zero to four. Their methods achieved an accuracy of only 38.1% at assigning the correct score, i.e., the score assigned by a human grader. However, as described in Section "Related Work", even this task is considerably different from ours. For example, essays have a strong spatial organization and a known lexicon, while the handwritten problem solutions we consider do not. Thus, given the complexities of our problem domain, a correlation of $r = .503$ between predicted and actual grades represents a reasonable level of performance.

Evaluating the validity and reliability of automated grading methods is a complicated matter. For example, Attali (2013) presents an analysis of the validity and reliability of AES methods. He notes that because these methods cannot evaluate the same aspects of writing that human graders do, many researchers evaluate the validity of AES methods simply in terms of their ability to match human-generated scores, without concern for which aspects of the writing the methods actually evaluate. We employ the same approach here. Understanding which aspects of problem solving our methods measure is an interesting and challenging question which is beyond the scope of our present work.

Similarly, we have not yet examined the reliability of our methods. We use the methods in (Stahovich and Lin 2016) to locate and recognize characters and locate equations groups. If these methods cannot interpret the writing, this will affect the computation of the lexical and pause features, and thus could affect the predicted grade. As result, two solutions that differ only in the legibility of the writing may be assigned different grades. Examining this issue is left to future work. Nevertheless, we believe that improving the accuracy of the underlying recognition methods we use will increase our accuracy at predicting grade.

We found that subsets of the features produced the strongest predictions of grade. We performed limited feature subset selection by enumerating all models containing three lexical features and the pause feature and selecting the best-performing ones. In future work, it will be necessary to employ more sophisticated subset selection techniques such as those in (Kohavi and John 1997).

Some of the lexical features are domain-independent, while others like the number of "Σ" characters may be specific to particular STEM subjects. Thus, future research is needed to determine if these results generalize to other STEM courses. Furthermore, replication of these results with other cohorts of students will strengthen the conclusions of this study.

Conclusion

This study demonstrated that the lexical properties of a student's handwritten solution to an exam problem in an engineering course correlate with the correctness of the work. We developed a set of 24 quantitative features characterizing the lexical properties of handwritten equations. These features include the number of occurrences of various classes of symbols, binary sequences of symbols, and tripartite sequences of symbols. We used these features to construct SVM regression models to predict the correctness of the work, i.e., the grade a human grader would assign.

We evaluated this approach on a dataset containing solutions to six exam problems from an undergraduate engineering course in statics. Students completed the exam problems using digital pens that recorded the work as time-stamped pen strokes. SVM regression models trained using the complete set of lexical features achieved a correlation of $r = .433$ ($p < .001$) on the six problems combined, and an average correlation of $r = .321$ for the problems considered individually.

We also examined the performance of our lexical features in combination with a pause feature that represents the number of long pauses in a student's handwritten solution (Stahovich and Lin 2016). We found that the two types of features provide complementary information about correctness and that combining the two produced the best performance. For example, SVM regression models trained using an optimized subset of three lexical features and the number of long pauses achieved an average correlation with grade across all six problems of $r = .503$. This is a surprising result given that our approach does not attempt to interpret the equations or even the final numerical answer. Additionally, unlike more traditional automated grading methods, such as automated exam scoring, our methods work from handwritten pen strokes rather than machine interpretable text.

One important property of our techniques is that they do not require complete semantic interpretation of equations, nor do they require knowledge of the subject matter. Consequently, our techniques should be readily extensible to other subject areas. In particular, we expect that our techniques will be useful for assessing student learning in a variety of STEM subjects.

Our techniques are an important step toward creating systems that can automatically grade handwritten coursework. While our current models cannot yet replace a human grader, our techniques are attractive because of their generality and low cost. By examining the steps used to solve a problem, our techniques complement traditional online homework systems that consider only the final answer.

Acknowledgements This material is based upon work supported by the National Science Foundation under Award Numbers 0935239, 1432820, and 1612511. Livescribe, Inc. provided some materials used in the project. We thank Daniel Jeske for his suggestions about some of the statistical analysis.

References

Attali, Y. (2013). Validity and reliability of automated essay scoring. Handbook of automated essay evaluation: current applications and new direction, 181–198.

Attali, Y. (2015). Reliability-based feature weighting for automated essay scoring. *Appl. Psychol. Meas.*, *39*(4), 303–313.

Beal, C.R., & Cohen, P.R. (2008). Temporal data mining for educational applications. In *Proceedings of the 10th Pacific rim international conference on artificial intelligence: trends in artificial intelligence* (pp. 66-77). Berlin: Springer.

Bransford, J.D., Brown, A.L., Cocking, R.R. (Eds.) (2000). *How people learn: brain, mind, experience, and school: expanded edition*. Washington: The National Academies Press.

Cheng, P.C., & Rojas-Anaya, H. (2008). Measuring mathematic formula writing competence: an application of graphical protocol analysis. In *Proceedings of the 13th annual conference of the cognitive science society* (pp. 869–874).

Demirci, N. (2010). Web-based vs. paper-based homework to evaluate students' performance in introductory physics courses and students' perceptions: two years experience. *Int. J. E-Learning*, *9*(1), 27–49.

Gikandi, J., Morrow, D., Davis, N. (2011). Online formative assessment in higher education: a review of the literature. *Comput. Educ.*, *57*(4), 2333–2351.

Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., Witten, I.H. (2009). The WEKA data mining software: an update. *ACM SIGKDD Explorations Newsletter*, *11*(1), 10–18.

Herold, J., & Stahovich, T. (2012). Characterizing students' handwritten self-explanations. In *Proceedings of the 2012 American society for engineering education annual conference and exposition*.

Herold, J., Stahovich, T.F., Rawsonm, K. (2013a). Using educational data mining to identify correlations between homework effort and performance. In *Proceedings of the 2013 American society for engineering education annual conference and exposition*.

Herold, J., Zundel, A., Stahovich, T.F. (2013b). Mining meaningful patterns from students' handwritten coursework. In *Proceedings of the sixth international conference on educational data mining*.

Kara, L.B., & Stahovich, T.F. (2005). An image-based, trainable symbol recognizer for hand-drawn sketches. *Comput. Graph.*, *29*, 501–517.

Kohavi, R., & John, G.H. (1997). Wrappers for feature subset selection. *Artif. Intell.*, *97*(1), 273–324. https://doi.org/10.1016/S0004-3702(97)00043-X, http://www.sciencedirect.com/science/article/pii/S000437029700043X, relevance.

Krüger, A., Merceron, A., Wolf, B. (2010). A data model to ease analysis and mining of educational data. In *Proceedings of the 3rd international conference on educational data mining*.

LaViola, J.J.J.r. (2007). An initial evaluation of mathpad^2: a tool for creating dynamic mathematical illustrations. *Comput. Graph.*, *31*(4), 540–553. https://doi.org/10.1016/j.cag.2007.04.008, https://doi.org/10.1016/j.cag.2007.04.008.

LaViola, J.J.J.r., & Zeleznik, R.C. (2004). Mathpad^2: a system for the creation and exploration of mathematical sketches. *ACM Trans. Graph.*, *23*(3), 432–440. https://doi.org/10.1145/1015706.1015741, http://doi.acm.org/10.1145/1015706.1015741.

Li, N., Cohen, W.W., Koedinger, K.R., Matsuda, N. (2011). A machine learning approach for automatic student model discovery. In *Proceedings of the 4th international conference on educational data mining* (pp. 31–40).

Mostow, J., Gonzàlez-Brenes, J.P., Tan, B.H. (2011). Learning classifiers from a relational database of tutor logs. In *Proceedings of the 4th international conference on educational data mining* (pp. 149–158).

Oviatt, S., Arthur, A., Cohen, J. (2006). Quiet interfaces that help students think. In *UIST '06: Proceedings of the 19th annual ACM symposium on User interface software and technology, ACM Press, New York* (pp. 191–200).

Pellegrino, J.W., Chudowsky, N., Glaser, R. (Eds.) (2001). *Knowing what students know: the science and design of educational assessment*. Washington: The National Academies Press.

Rabiner, L. (1989). A tutorial on hidden markov models and selected applications in speech recognition. *Proc. IEEE*, *77*(2), 257–286.

Rawson, K., & Stahovich, T.F. (2013). Predicting course performance from homework habits. In *Proceedings of the 2013 American society for engineering education annual conference and exposition*.

Rawson, K., Stahovich, T.F., Mayer, R.E. (2017). Homework and achievement: using smartpen technology to find the connection. *J. Educ. Psychol.*, *109*(2), 208.

Romero, C., & Ventura, S. (2010). Educational data mining: a review of the state of the art. *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, *40*(6), 601–618.

Romero, C., Romero, J., Luna, J., Ventura, S. (2010). Mining rare association rules from e-learning data. Educ. Data Mining, 171–180.

Schneider, S.C. (2014). Paperless grading of handwritten homework: electronic process and assessment. In *Proceedings of the American society for enginering education annual conference*.

Shanabrook, D.H., Cooper, D.G., Woolf, B.P., Arroyo, I. (2010). Identifying high-level student behavior using sequence-based motif discovery. In de Baker, R.S.J., Merceron, A., Jr, P.I.P. (Eds.) *Proceedings of the 3rd international conference on educational data mining* (pp. 191–200).

Sharma, A., & Jayagopi, D.B. (2018). Automated grading of handwritten essays. In *2018 16Th international conference on frontiers in handwriting recognition (ICFHR), IEEE* (pp. 279–284).

de Silva, R., Bischel, D.T., Lee, W., Peterson, E.J., Calfee, R.C., Stahovich, T.F. (2007). Kirchhoff's pen: a pen-based circuit analysis tutor. In *Proceedings of the 4th eurographics workshop on sketch-based interfaces and modeling, ACM, New York, NY, USA, SBIM '07* (pp. 75–82).

Singh, A., Karayev, S., Gutowski, K., Abbeel, P. (2017). Gradescope: a fast, flexible, and fair system for scalable assessment of handwritten work. In *Proceedings of fourth ACM conference on learning@ scale* (pp. 81–88): ACM.

Smithies, S., Novins, K., Arvo, J. (1999). A handwriting-based equation editor. In *Proceedings of the 1999 conference on graphics interface '99, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA* (pp. 84–91). http://dl.acm.org/citation.cfm?id=351631.351660.

Srihari, S.N., Srihari, R.K., Babu, P., Srinivasan, H. (2007). On the automatic scoring of handwritten essays. In *Proceeddings of international joint conference on artificial intelligence* (pp. 2880–2884).

Stahovich, T.F., & Lin, H. (2016). Enabling data mining of handwritten coursework. *Comput. Graph.*, *57*, 31–45. https://doi.org/10.1016/j.cag.2016.01.002, http://www.sciencedirect.com/science/article/pii/S0097849316300012.

Steif, P.S., & Dollár, A. (2009). Study of usage patterns and learning gains in a web-based interactive static course. *J. Eng. Educ.*, *98*(4), 321–333.

Steif, P.S., Lobue, J.M., Kara, L.B., Fay, A.L. (2010). Improving problem solving performance by inducing talk about salient problem features. *J. Eng. Educ.*, *99*(2), 135–142.

Stevens, R., Johnson, D.F., Soller, A. (2005). Probabilities and predictions: modeling the development of scientific problem-solving skills. *Cell Biol. Educ.*, *4*(1), 42–57.

Trivedi, S., Pardos, Z.A., Sàrközy, G.N., Heffernan, N.T. (2011). Spectral clustering in educational data mining. In *Proceedings of the 4th international conference on educational data mining* (pp. 129–138).

Van Arsdale, T., & Stahovich, T. (2012). Does neatness count? What the organization of student work says about understanding. In *Proceedings of the 2012 American society for engineering education annual conference and exposition*.