ABSTRACT
Curricula often lack metadata to characterize the relatedness of concepts. To investigate automatic methods for generating relatedness metadata for a mathematics curriculum, we first address the task of identifying which terms in the vocabulary from mathematics word problems are associated with the curriculum. High chance-adjusted interannotator agreement on manual identification of math terms was achieved by considering terms in their contexts. These terms represent 13% of the vocabulary in one seventh grade mathematics text. Six classification algorithms were compared to classify math terms for this text. To avoid overfitting to this curriculum, we relied on a small number of features that exploit external knowledge sources.

1. INTRODUCTION
Curricula often lack metadata to characterize the relatedness of concepts. Our ultimate goal is to develop methods for automatic generation of knowledge graphs for mathematics from existing curricula. Towards that end, we develop a representation for math word problems that allows us to measure similarities between problems, based on the math terminology they share [14]. In this paper, we present our methods to automatically identify the math terms. While mathematics is a highly structured domain with many sources that define terms, we found no single source that captured the mathematics terms as used in the context of this curriculum. Furthermore, several terms that occur in the word problems, such as independent, chances, and set, are polysemous, but occur more frequently in a “mathematical” sense. We therefore annotated the full vocabulary as “math” or “non-math” based on the predominant usage in the curriculum, and found high agreement among annotators. We then tested six methods for automatic classification.

2. RELATED WORK
Adaptive learning environments (ALEs) have shown promising results for mathematics and other STEM subjects [18, 5, 1], even when compared with human tutors [24]. For ALE’s, the domain model is typically created anew but automated methods have been applied [3] [25]. The latter build concept maps from handbooks about SCORM standards, based on hand-constructed patterns to match dependency parses, then use the concept maps to build ontologies. Our work also derives semantic knowledge from text, aimed at representing semantic relations among mathematics word problems.

Automated methods have also been used in construction of educational domain models for assessments [20], standards [9], and targeted prerequisites for learners [13]. Various approaches have been used to represent domain knowledge, including semantic networks with frames and production rules [23], or model-tracing architectures to identify problem-solving steps students take, including incorrect ones [2]. Model-tracing, inherently reactive, has been extended with tutorial actions to pro-actively guide students [12]. Other approaches to automatically generate metadata require existing domain ontologies [22]. Our goal is to develop a network of relations among problems that could be used pro-actively by ALEs or teachers to move students through the curriculum in a way that promotes optimal learning.

To represent mathematics word problems, we create a bag-of-words (BOW) vector for math words using methods similar to terminology identification [10]. In separate work, we use this vector to create similarity networks among problems [14]. A range of methods have been used to identify terms in product reviews [6], concepts in semi-structured data [4], technical language in patents [15], or domain-specific terminology in general [21]. Much of this work deals with identi-
The table shows a proportional relationship between \( x \) and \( y \). Complete the table.

Solve the inequality \( x + 1 < 4 \). Then graph the solutions.

Figure 1: Sample word problems.

Data: Mathematics Exercises

The data consists of 3000 word problems from a Grade 7 mathematics curriculum. The problems, which can incorporate images, tables, and graphs, are instantiated through templates. Figure 1 shows two problem exercises from chapters 2 and 9, with words that evoke math concepts in boldface. Note that a template, \( x\{+|−\}X\{<|>\}Y \), randomly generates instances such as \( x + 4 > 9 \) or \( x + 1 < 4 \). Depending on the number of instance variables and constraints, a template may generate a bounded or nearly limitless number of instances. In addition to the exercise itself, which may contain a few steps that are typically solved via multiple choice or fill-in-the-blank, learners are able to select a more detailed guided solution, or to view the steps to solve a sample problem instance. We created an XML parser to extract the text from the exercises, the guided solutions, and sample problems. The vocabulary analysis is based on the extracted text.

Annotation and Reliability

At 4,495 words (not lemmatized), the curriculum’s vocabulary is relatively small. Removal of typical stopwords leaves 4,283 words. An additional 103 words, while not typical stop words, have very high frequency across problems (e.g., amount, answer, compare) and are not likely to be useful for measuring semantic similarity among problems.

The terms we are interested in are those that are characteristic of the concepts the students should know to demonstrate mastery of the curriculum. The three co-authors, working independently, each labeled an initial sample of 100 words as math, non-math and other, based on initial guidelines. Because pairwise agreement can be high when a chance-adjusted agreement coefficient is low (the so-called paradox of kappa [11]), agreement was measured using both pairwise agreement and Krippendorff’s Alpha [16], a metric that factors out chance agreement. Initially, pairwise agreement was 0.93, but Alpha was 0.54, which is rather low. The low chance-adjusted agreement was mainly due to inconsistency among annotators in looking at the contexts in which words were used, and also due to borderline cases. We wrote more explicit guidelines with examples (4 pages), then labeled two additional samples of 100 words each, computing agreement on each sample before proceeding to the next. On the second and third samples, pairwise agreement was 0.92 in both cases, and Alpha was 0.83 and 0.81. Given the high agreement and consistency across the second and third samples, we determined the labeling to be reliable. One of the co-authors labeled the remainder of the vocabulary, yielding 3832 words labeled as non-math, 571 as math and 92 as other. Only the words labeled as math and non-math were used to train the classifier.

Classification Experiments

This section reports results from a suite of classification algorithms applied to the labeled data. To represent the vocabulary for the learner, we engineered features based on search and glossary information, and on a corpus-based metric. Two challenges for the classification were infrequency of the positive class (high data skew), and apparent non-linearity of the class separation. Of six learning algorithms, those that had best performance were most suited to these learning challenges, as described further below.

5.1 Feature Representation

We constructed a feature vector representation for the words with the 6 features listed in Figure 2. All feature values were scaled to be in the range of 0 to 1.

For the first two features listed in Figure 2, we used the functionality of Google Custom Search that permits customized searches to user-specified domains. For the first feature we queried mathworld.wolfram.com, and for the second we queried math.about.com. The value for each of these features consists of the total number of query returns, which can be arbitrarily large.

Google Custom Search can also be configured so that for the top ten returns to a query, each return consists of a triple with the url, a list of text snippets containing the term at that url, and the page title at that url. For the third feature we queried mathworld.wolfram.com, and for the second we queried math.about.com. The value for each of these features consists of the total number of query returns, which can be arbitrarily large.

Bing is a Microsoft search engine with an interface through which queries can be made programmatically. The interface returns the top 50 search results. Like Google searches, each result contains the relevant URL, snippets, and title of the page. As in the Google search feature, for the fourth feature in Figure 2, a counter was incremented whenever math,
mathematics, or arithmetic occurred at least once in a triple element. Values are in \([0,150]\).

The mathematics curriculum has an associated glossary of 246 math terms. It includes simple terms, e.g., “sphere,” and compound terms, e.g., “associative property of multiplication.” The glossary was expanded with the individual words in compound terms, excluding stop words. Thus for the compound term “associative property of multiplication”, the words associative, property and multiplication were added. In this way, the glossary was expanded to 516 terms. A boolean feature value was used here to indicate exact occurrence of a word in the glossary.

Relative frequency ratio (RFR) measures relative frequency of a term in reference to a contrastive background corpus \([6, 8]\). The frequency of a word \(w_i\) in a corpus \(C\), expressed as \(FR(w_i, C)\), is its count normalized by size of the corpus. For a domain specific corpus, e.g., a mathematics text, the frequency of domain-specific terms should be higher than in a large, background corpus. The formula for RFR is:

\[
RFR(w_i) = \frac{FR(w_i, DC)}{FR(w_i, BC)}
\]

where \(DC\) is the domain corpus and \(BC\) is the background corpus. We tested RFR with two background corpora: the Open American National Corpus (OANC; \(N=22 \times 10^6\)) and English Gigaword, Fifth Edition (\(N=4.033 \times 10^6\)). Unsurprisingly, we found that the size of the background corpus is critical to the precision of the RFR measures. When we ranked Digits words by RFR scores using Gigaword, 306 of the words labeled as “math” occur in the top 1,000 words compared with 248 using OANC. Therefore we used Gigaword as the background corpus.

### 5.2 Classification

The labeled data was randomly split into a training set with 75% of the vocabulary (3301 terms) and a test set with 25% of the vocabulary (1101 terms). Using logistic regression, classification results yielded an overall precision of 0.87 and a recall of 0.88, compared with 0.78 precision and 0.25 recall for the math class. The low recall of math terms can be attributed to high class imbalance, where only 13% of terms are in the math class. Linear SVM also yielded poor results, suggesting that the classes cannot be linearly separated. To address the class imbalance, we use class weights for SVM, where we use polynomial and RBF kernels to address the non-linearity. Ensembles of weak learners also help with the class imbalance, where only 13% of terms are in the math class. Linear SVM also yielded poor results, suggesting that the classes cannot be linearly separated.

### 6. RESULTS AND DISCUSSION

Table 1 shows the results for the six classification experiments. All the classifiers had high accuracy, due to the high class imbalance favoring non-math words. Accuracy on the math words (sensitivity), however, was relatively low for all but the SVM learners. The ensemble methods had higher precision on the math words (\(\geq 0.78\)) but low sensitivity (0.41-0.46). The SVM learners had lower precision (about 0.5) and higher sensitivity (0.68). The logistic regression had high precision on the math words (0.81) but very low sensitivity (0.32). For g-mean, all the classifiers had values above 0.50, indicating respectable performance. The two SVM learners, however, had the highest g-means: 0.78 (polynomial kernel) and 0.79 (RBF kernel).

Manual error analysis of math words that were incorrectly classified by multiple learners indicated that many of the errors were due to polysemous words that have one or more non-math senses that occur with non-negligible frequency. This includes words like point, dependent, and trial. In WordNet [19], for example, point used as a noun has twenty-five senses, and fourteen senses used as a verb. Future work on the classification task will include investigation of features commonly used for coarse-grained word sense disambiguation, where accuracies of 88% have been achieved using lexical, syntactic and topical features [7] so that we can apply the same methods to new curricula.

### 7. CONCLUSIONS

The vocabulary classification task we address, to identify vocabulary that characterizes the semantics of a curriculum, differs from standard terminology detection, where the focus is on highly technical compound terms. It also differs from word sense disambiguation in that we are interested in binary classification of senses, based on the use of terms for a given curriculum. We have shown that human annotators can achieve very high pairwise and chance-adjusted agreement. To avoid overfitting to a given curriculum, the features we used draw on external knowledge sources such as glossaries, web search and large background corpora. With relatively few such features and choice of an appropriate learning

### Table 1: Classification Results

<table>
<thead>
<tr>
<th>Classifier</th>
<th>Precision</th>
<th>Recall</th>
<th>F-score</th>
<th>Sensitivity</th>
<th>Specificity</th>
<th>G-Mean</th>
</tr>
</thead>
<tbody>
<tr>
<td>adaboost</td>
<td>0.89</td>
<td>0.90</td>
<td>0.89</td>
<td>0.43</td>
<td>0.97</td>
<td>0.64</td>
</tr>
<tr>
<td>bagging</td>
<td>0.90</td>
<td>0.91</td>
<td>0.90</td>
<td>0.41</td>
<td>0.98</td>
<td>0.63</td>
</tr>
<tr>
<td>rand-forest</td>
<td>0.90</td>
<td>0.91</td>
<td>0.90</td>
<td>0.45</td>
<td>0.97</td>
<td>0.66</td>
</tr>
<tr>
<td>SVM-poly</td>
<td>0.89</td>
<td>0.86</td>
<td>0.87</td>
<td>0.68</td>
<td>0.89</td>
<td>0.78</td>
</tr>
<tr>
<td>SVM-RBF</td>
<td>0.89</td>
<td>0.87</td>
<td>0.88</td>
<td>0.68</td>
<td>0.90</td>
<td>0.79</td>
</tr>
<tr>
<td>logistic regression</td>
<td>0.89</td>
<td>0.90</td>
<td>0.88</td>
<td>0.31</td>
<td>0.98</td>
<td>0.56</td>
</tr>
</tbody>
</table>
algorithm, we achieve very high accuracy and good sensitivity, despite the small proportion of the positive class.

8. ACKNOWLEDGMENTS

We would like to thank the Research and Innovation Network at Pearson for support of this work.

9. REFERENCES


