Semantic Similarity Graphs of Mathematics Word Problems: Can Terminology Detection Help?

Rogers Jeffrey Leo John
Center for Computational Learning Systems
Columbia University
New York, NY, USA
rl2689@columbia.edu

Rebecca J. Passonneau
Center for Computational Learning Systems
Columbia University
New York, NY, USA
becky@ccls.columbia.edu

Thomas S. McTavish
Center for Digital Data, Analytics & Adaptive Learning
Pearson
Austin, TX, USA
tom.mctavish@pearson.com

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.

The vocabulary items to be classified were represented using a small number of features based on glossaries, web search, and corpus statistics. Only 13% of the terms in our vocabulary were labeled as "math." Such data skew is challenging for many machine learning methods. To address the class imbalance, we used ensembles of weak learners and support vector machines (SVMs), weighting errors on the "math" class more heavily. We found that SVMs were our best classifiers. The automated methods presented here can enhance existing math curricula with domain knowledge graphs of content similarity among word problems.

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 proactively 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-

Chap.	Sec.	Exer.	Text
2	1	19	The table shows a **proportional** relationship between x and y. Complete the table.
9	1	11	Solve the **inequality** x + 1 < 4. Then graph the solutions.

Figure 1: Sample word problems.

fication of multi-word noun-noun compounds of a technical nature, and ranking them. In contrast, the secondary school math terminology has few compounds, includes a mix of different parts of speech, and is non-technical. As in [21, 6, 4], we rely on relative frequency ratio [8] to distinguish the frequencies of words in our corpus from their frequencies in a large background corpus. Unlike most of this work, apart from [15], we developed annotation guidelines and measured interannotator agreement. We find an agreement of 0.81 among three annotators using Krippendorff's α (see below), compared to 0.76 (Fleiss's κ; a similar metric) in [15].

3. 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, $x\{+|-\}X\{< \mid >\}Y$, 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.

4. 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.

1. Wolfram Mathworld
2. About.com: mathematics
3. Math domains in Google search results
4. Math domains in Bing search results
5. Digits math glossary
6. Relative frequency ratio

Figure 2: Features to represent vocabulary

5. 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 listed in Figure 2, we query the web using this functionality, and calculate the feature value based on the triples for the top ten returns. Each time *math*, *mathematics*, or *arithmetic* occurs at least once in each element of a triple, a counter is incremented. The maximum value is thus 30.

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*,

Table 1: Classification Results

Classifier	Precision	Recall	Fscore	Sensitivity	Specificity	G-Mean
adaboost	0.89	0.90	0.89	0.42	0.97	0.64
bagging	**0.90**	**0.91**	**0.90**	0.41	**0.98**	0.63
rand-forest	**0.90**	**0.91**	**0.90**	0.45	0.97	0.66
SVM-poly	0.89	0.86	0.87	**0.68**	0.89	**0.78**
SVM-RBF	0.89	0.87	0.88	**0.68**	0.90	**0.79**
logistic regression	0.89	0.90	0.88	0.31	**0.98**	0.56

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:

$$RFR(w_i) = \frac{FR(w_i, DC)}{FR(w_i, BC)} \quad (1)$$

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 non-linearity. For each of three ensemble methods, *Boosting*, *Bagging* and *Random Forests*, we used 1000 *Decision Trees*.

Evaluation results are reported using precision, recall, f-measure, and g-mean [17]. The latter, the geometric mean of accuracy on the positive class (recall, or sensitivity) and accuracy on the negative class (specificity), is high when both accuracies are high and their difference is small. It is particularly useful when there are no criteria for constructing a cost matrix for errors in sensitivity versus specificity.

For the SVM classifiers, we used C=10,000. For the polynomial kernel, the degree was 4 and the class weights assigned to the math and non-math classes were 270 and 1350 respectively. For the SVM with the RBF Kernel, class weights were set to 200 and 1100.

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 very 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 peformance. 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-neglible 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-adusted agreement. To avoid overfitting to a given curriculm, 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

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

[1] V. Aleven, A. Ogan, O. Popescu, C. Torrey, and K. R. Koedinger. Evaluating the effectiveness of a tutorial dialogue system for self-explanation. In J. Lester, R. M. Vicari, and F. Paraguaca, editors, *Intelligent Tutoring Systems: Seventh International Conference, ITS 2004*, pages 443–454. Springer, 2004.

[2] J. R. Anderson and R. Pelletier. A developmental system for model-tracing tutors. In L. Birnbaum, editor, *The International Conference on the Learning Sciences*, pages 1–8. Association for the Advancement of Computing in Education, Charlottesville, VA, 1991.

[3] L. Aroyo, P. Dolog, G.-J. Houben, M. Kravcik, A. Naeve, M. Nilsson, F. Wild, and others. Interoperability in personalized adaptive learning. *Educational Technology & Society*, 9(2):4–18, 2006.

[4] T. Atapattu, K. Falkner, and N. Falkner. Automated extraction of semantic concepts from semi-structured data: Supporting computer-based education through the analysis of lecture notes. In S. W. Liddle, K.-D. Schewe, A. M. Tjoa, and X. Zhou, editors, *Database and Expert Systems Applications*, number 7446 in Lecture Notes in Computer Science, pages 161–175. Springer Berlin Heidelberg, 2012.

[5] C. R. Beal, I. M. Arroyo, P. R. Cohen, and B. P. Woolf. Evaluation of AnimalWatch: An intelligent tutoring system for arithmetic and fractions. *Journal of Interactive Online Learning*, 9(1):64–77, 2010.

[6] J. Broß and H. Ehrig. Terminology extraction approaches for product aspect detection in customer reviews. In *Proceedings of the Seventeenth Conference on Computational Natural Language Learning*, pages 222–230, Sofia, Bulgaria, August 2013. Association for Computational Linguistics.

[7] J. F. Cai, W. S. Lee, and Y. W. Teh. NUS-ML: Improving word sense disambiguation using topic features. In *Proceedings of the 4th International Workshop on Semantic Evaluations (SemEval 07)*, pages 249–252, Stroudsburg, PA, USA, 2007. Association for Computational Linguistics.

[8] F. J. Damerau. Generating and evaluating domain-oriented multi-word terms from texts. *Information Processing and Management*, 29(4):433–447, 1993.

[9] H. Devaul, A. R. Diekema, and J. Ostwald. Computer-assisted assignment of educational standards using natural language processing. *Journal of the American Society for Information Science and Technology*, 62(2):395–405, 2011.

[10] P. Drouin, N. Grabar, T. Hamon, and K. Kageura, editors. *Proceedings of the 4th International Workshop on Computational Terminology (Computerm)*. Association for Computational Linguistics and Dublin City University, Dublin, Ireland, August 2014.

[11] A. R. Feinstein and D. V. Cicchetti. High agreement but low Kappa: I. The problems of two paradoxes. *Journal of Clinical Epidemiology*, 43(6):543–549, 1990.

[12] N. T. Heffernan, K. R. Koedinger, and L. Razzaq. Expanding the model-tracing architecture: A 3rd generation intelligent tutor for algebra symbolization. *International Journal of Artificial Intelligence in Education*, 18:153–178, 2008.

[13] S. Jain and J. Pareek. Automatic extraction of prerequisites and learning outcome from learning material. *International Journal of Metadata, Semantics and Ontologies*, 8(2):145–154, Jan. 2013.

[14] R. J. L. John, T. S. McTavish, and R. J. Passonneau. Semantic graphs for mathematics word problems based on mathematics terminology. In *WS-1: Graph-based Educational Data Mining (G-EDM 2015)*, 2015.

[15] A. Judea, H. Schütze, and S. Bruegmann. Unsupervised training set generation for automatic acquisition of technical terminology in patents. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 290–300, Dublin, Ireland, August 2014. Dublin City University and Association for Computational Linguistics.

[16] K. Krippendorff. *Content analysis*. Sage Publications, Beverly Hills, CA, 1980.

[17] M. Kubat, Robert, and S. Matwin. When negative examples abound. In *Proceedings of the 9th European Conference on Machine Learning*, ECML '97, pages 146–153, London, UK, UK, 1997. Springer-Verlag.

[18] E. Melis, E. Andres, J. Budenbender, A. Frischauf, G. Goduadze, P. Libbrecht, M. Pollet, and C. Ullrich. Activemath: A generic and adaptive web-based learning environment. *International Journal of Artificial Intelligence in Education (IJAIED)*, 12:385–407, 2001.

[19] G. A. Miller. WordNet: A lexical database for English. *Commun. ACM*, 38(11):39–41, Nov. 1995.

[20] R. J. Mislevy, J. T. Behrens, K. E. Dicerbo, and R. Levy. Design and discovery in educational assessment: Evidence-centered design, psychometrics, and educational data mining. *JEDM - Journal of Educational Data Mining*, 4(1):11–48, Oct. 2012.

[21] Y. Park, R. J. Byrd, and B. K. Boguraev. Automatic glossary extraction: Beyond terminology identification. In *Proceedings of the 19th International Conference on Computational Linguistics - Volume 1*, COLING '02, pages 1–7, Stroudsburg, PA, USA, 2002. Association for Computational Linguistics.

[22] D. Roy, S. Sarkar, and S. Ghose. Automatic extraction of pedagogic metadata from learning content. *Int. J. Artif. Intell. Ed.*, 18(2):97–118, Apr. 2008.

[23] S. Stankov, M. Rosić, Žitko, and A. Grubišić. TEx-Sys model for building intelligent tutoring systems. *Computers and Education*, 51:1017–1036, 2008.

[24] K. VanLehn. The relative effectiveness of human tutoring, intelligent tutoring systems, and other tutoring systems. *Educational Psychologist*, 46(4):197–221, 2011.

[25] A. Zouaq and R. Nkambou. Building domain ontologies from text for educational purposes. *IEEE Transactions on Learning Technologies*, 1(1):49–62, Jan. 2008.