Semantic Graphs for Mathematics Word Problems based on Mathematics Terminology

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

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

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

ABSTRACT

We present a graph-based approach to discover and extend semantic relationships found in a mathematics curriculum to more general network structures that can illuminate relationships within the instructional material. Using words representative of a secondary level mathematics curriculum we identified in separate work, we constructed two similarity networks of word problems in a mathematics textbook, and used analogous random walks over the two networks to discover patterns. The two graph walks provide similar global views of problem similarity within and across chapters, but are affected differently by number of math words in a problem and math word frequency.

1. INTRODUCTION

Curricula are compiled learning objects, typically presented in sequential order and arranged hierarchically, as in a book's Table of Contents. Ideally, a *domain model* captures relationships between the learning objects and the knowledge components or skills they exercise. Unfortunately, domain models are not often granular enough for optimal learning experiences. For example, prerequisite relationships may be lacking, or the knowledge components associated with an exercise may be unknown. In such cases, assessments on those learning objects will be insufficient to enable appropriate redirection unless expert (i.e. teacher) intervention is explicitly given. Domain models remain coarse because using experts to enumerate and relate the knowledge components is costly.

As a means to automatically discover relationships among learning objects and to reveal their knowledge components, we demonstrate the use of direct similarity metrics and random graph walks to relate exercises in a mathematics curriculum. We first apply a standard cosine similarity measure between pairs of exercises, based on bag-of-word vectors consisting of math terms that we identified in separate work [7]. Then, to extract less explicit relationships between exercises, we randomly walk a graph using the cosine distance as edge weights. We also recast the problem as a bipartite graph with exercises on one side and words on the other, providing an edge when an exercise contains the math word. We contrast these two different types of random walks and find somewhat similar results, which lends confidence to the analysis. The bipartite graph walks, however, are more sensitive to differences in word frequency. Casting measures of similarity as graphs and performing random walks on them affords more nuanced ways of relating objects, which can be used to build more granular domain models for analysis of prerequisites, instructional design, and adaptive learning.

2. RELATED WORK

Random walks over graphs have been used extensively to measure text similarity. Applications include similarity of web pages [15] and other documents [5], citations [1], passages [14], person names in email [12] and so on. More recently, general methods that link graph walks with external resources like WordNet have been developed to produce a single system that handles semantic similarity for words, sentences or text [16]. Very little work compares walks over graphs of the same content, where the graphs have different structure. We create two different kinds of graphs for mathematics word problem and compare the results. We find that the global results are very similar, which is good evidence for the general approach, and we find differences in detail that suggest further investigation could lead to customizable methods, depending on needs.

An initiative where elementary science and math tests are a driver for artificial intelligence has led to work on knowledge extraction from textbooks. Berant et al. [2] create a system to perform domain-specific deep semantic analysis of a 48 paragraphs from a biology textbook for question answering. Extracted relations serve as a knowledge base against which to answer questions, and answering a question is treated as finding a proof. A shallow approach to knowledge extraction from a fourth grade science curriculum is taken in [6], and the knowledge base is extended through dialog with users until a path in the knowledge network can be found that supports a known answer. In the math domain, Kushman et al. [10] generate a global representation of algebra problems in order to solve them by extracting relations from sentences and aligning them. Seo et al. [18] study text and diagrams together in order to understand the diagrams better through textual cues. We are concerned with alignment of content

Two machines produce the same type of widget. Machine A produces W widgets, X of which are damaged. Machine B produces Y widgets, Z of which are damaged. The **fraction** of damaged widgets for Machine A is $\frac{X}{W}$ or (*simplified fraction*). The **fraction** of damaged widgets for Machine B is $\frac{Z}{Y}$ or (*simplified fraction*). Write each **fraction** as a **decimal** and a **percent**. Use pencil and paper. Select a small **percent** that would allow for a small **number** of damaged widgets. Find the **number** of widgets by which each machine exceeded the acceptable **number** of widgets.

Figure 1: Sample problem; math terms are in boldface.

across rather than within problems, and our objective is finer-grained analysis of curricula.

Other work that addresses knowledge representation from text includes ontology learning [3], which often focuses on the acquisition of sets of facts from text [4]. There has been some work on linking lexical resources like WordNet or FrameNet to formal ontologies [17, 13], which could provide a foundation for reasoning over facts extracted from text. We find one work that applies relation mining to e-learning: Šimko and Bieliková [19] apply automated relation mining to extract relations to support e-course authoring in the domain of teaching functional programming. Li et al. [11] apply k-means clustering to a combination of problem features and student performance features, and propose the clusters correspond to Knowledge Components [8].

3. METHODS

3.1 Data

We used 1800 exercises from 17 chapters of a Grade 7 mathematics curriculum. Most are word problems, as illustrated in Figure 1. They can incorporate images, tables, and graphs, but for our analysis, we use only the text. The vocabulary of the resulting text consists of 3,500 distinct words. We construct graphs where math exercises are the nodes, or in a bipartite graph, math exercises are the left side nodes and words are the right side nodes. Our initial focus is on exercise similarity due to similarity of the math skills that exercises tap into, and we use mathematics terminology as an indirect proxy of skills a problem draws upon.

3.2 Math Terminology

The text of the word problems includes ordinary language expressions unrelated to the mathematics curriculum, such as the nouns *machines, widgets* shown in problem in Figure 1, or the verbs *produces, damaged*. For our purposes, mathematics terminology consists of words that expresses concepts that are needed for the mathematical competence the curriculum addresses. To identify these terms, we developed annotation guidelines for human annotators who label words in their contexts of use, and assessed the reliability of annotation by these guidelines. Words can be used in the math texts sometimes in a math sense and sometimes in a non-math sense. Annotators were instructed to label terms based on the most frequent usage.

Using a chance-adjusted agreement coefficient in [-1,1] [9], reliability among three annotators was 0.81, representing high agreement. All the non-stop words were then labeled by a trained annotator. We developed a supervised machine learning approach to classify vocabulary into math and non-math words [7] that can be applied to new mathematics curricula. For the text used here, there were 577 math terms.

3.3 Random Walks in Graphs

A random walk on a graph starts at a given node and steps with random probability to a neighboring node. The same random decision process is employed at this and every subsequent node until a termination criterion is met. Each time a node is visited, it is counted. Open random walks require that the start node and end nodes differ. Traversal methods may employ a bias to navigate toward or away from certain neighbors through edge weights or other graph attributes.

In a graph, $G = (V, E)$ with nodes V and edges E, a random walk that begins at v_x and ends at v_y can be denoted as $(v_x, ..., v_y)$. By performing several random walks, the fraction of times the node v_y is visited converges to the probability of target v_y being visited given the start node v_x, which can be expressed as $P(v_y|v_x)$ under the conditions of the walk. In the case of a random walk length of 1, $P(v_y|v_x)$ will simply measure the probability of v_y being selected as an adjacent node to v_x.

3.4 Cosine Similarity Graph

Math exercises are represented as bag-of-words vectors with boolean values to indicate whether a given math term is present. Cosine similarity quantifies the angle between the two vectors, and is given by the dot product of two vectors.

$$\cos(\mathbf{t}, \mathbf{e}) = \frac{\mathbf{te}}{\|\mathbf{t}\|\|\mathbf{e}\|} = \frac{\sum_{i=1}^{n} \mathbf{t}_i \mathbf{e}_i}{\sqrt{\sum_{i=1}^{n} (\mathbf{t}_i)^2} \sqrt{\sum_{i=1}^{n} (\mathbf{e}_i)^2}} \quad (1)$$

Similarity values of 1 indicate that both the vectors are the same whereas a value of zero indicates orthogonality between the two vectors. Pairwise cosine similarities for all 1800 exercises were computed, yielding a cosine similarity matrix M_{cos}. The matrix corresponds to a graph where non-zero cosine similarities are edge weights between exercises.

In a graph walk, the probability that a node v_y will be reached in one step from a node v_x is given by the product of the degree centrality of v_x and the normalized edge weight (v_x, v_y). With each exercise as a starting node, we performed 100,000 random walks on the cosine-similarity graph, stepping with proportional probability to all outgoing cosine similarity weights. To measure 2nd degrees of separation, with each walk we made two steps.

For two math vectors considered as the sets A and B, cosine similarity can be conceptualized in terms of the intersection set $C = A \cup B$ and set differences $A \setminus B$ and $B \setminus A$. Cosine similarity is high when $|C| \gg A \setminus B$ and $|C| \gg B \setminus A$.

The degree of a node affects the probability of traversing any edge from that node. The two factors that affect degree centrality of a start node are the document frequencies of its math words, and the total number of math words. Here, document frequency (df) is the normalized number of exercises a word occurs in. A high df math word in a problem increase its degree centrality because there will be more

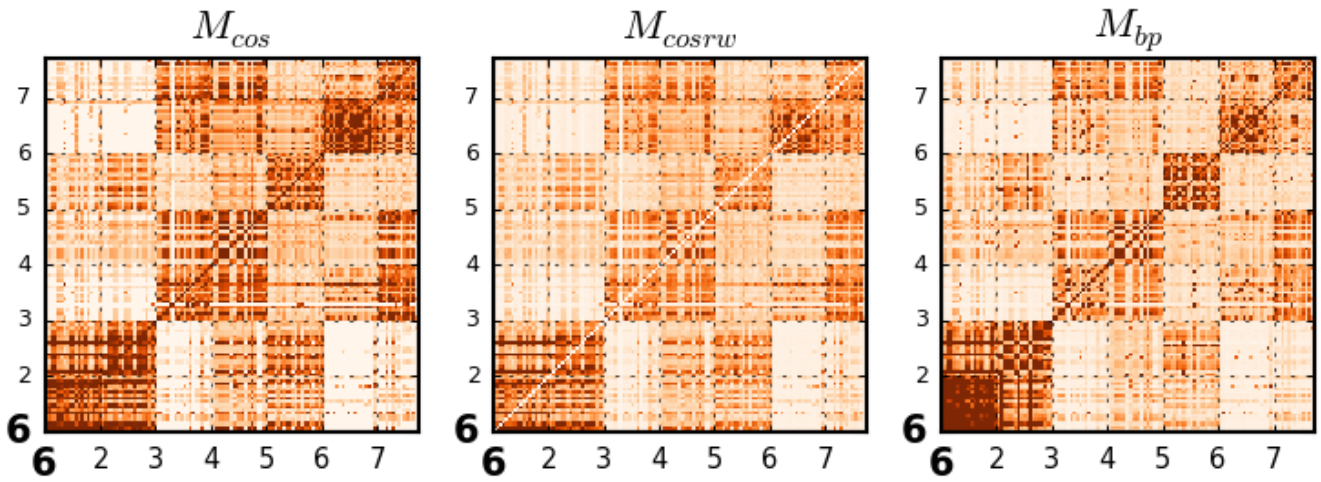

Figure 2: Exercise-to-Exercise similarity in Chapter 6. The exercises of Chapter 6 are displayed in row-columns in a square matrix. The rows represent source nodes and columns represent targets. Each row has been normalized across the book, even though only Chapter 6 is shown. The axes demarcate the sections of the chapter. M_{cos} is the cosine similarity. M_{cosrw} is the output from the random walk using cosine similarity edge weights, M_{bp} is the output from the random bipartite walk. Raw values displayed between 0 and 0.005 corresponding to light and dark pixels, respectively.

problems it can share words with, resulting in non-zero cosine values and therefore edges. The number of math words in a problem also increases its degree centrality.

3.5 Bipartite exercise and word graph

The set of exercises V_e are the left-side nodes and the math words V_w are the right-side nodes in the undirected bipartite graph $G = (V_e, V_w, E)$, where an edge exists between v_{ex} and v_{wi} if exercise x contains the math word i.

We performed open random walks on this graph to measure similarity between nodes. To measure the similarity of exercises, we walk in even steps – a step to a connected word followed by a step back to one of the exercises that shares that word. The degrees of separation between vertices on the same side of the graph (e.g. exercise-to-exercise) will be $l/2$ where l is the length of the walk. In this paper, we explored first and second degrees of separation so our bipartite graphs had a walk length of 4.

Table 1: Summary statistics of the similarity distributions

	cosine	rw_{cos}	rw_{bp}
minimum	0	0	0
maximum	6.3×10^{-2}	0.50	0.11
mean	5.55×10^{-4}	5.55×10^{-4}	5.55×10^{-4}
median	0	2.41×10^{-4}	2.06×10^{-4}
std. dev.	1.19×10^{-3}	8.57×10^{-4}	1.24×10^{-3}

Because exercise nodes are connected via word nodes, we interpret the fraction of node visits as a similarity measure between the source node and any node visited. We performed 100,000 random walks from each node. Exercise-to-exercise similarity can be visualzed as square matrices with source nodes in the rows and target nodes in the columns. To factor out the times a source may have been selected as one of the targets, we set the diagonal of the matrix to zero. We then normalized across the rows so that we could interpret the distribution across the row as a probability distribution to all other nodes for that source node.

4. RESULTS

We compare the three measures of similarity between exercises: 1) cosine similarity, 2) random walks using cosine similarity as edge weights, and 3) random walks along a bipartite graph of exercises and words.

4.1 Exercise-to-Exercise Similarity

We describe exercise-to-exercise similarity with square matrices where each exercise is represented as a row-column. A number of features of the measures are embedded in Figure 2, which shows heatmaps of color values for pairs of exercises in chapter 6 for each matrix. We find that within chapters and especially within sections of those chapters, there is a high degree of similarity between exercises regardless of the measure. This demonstrates that words within sections and chapters share a common vocabulary. We can see that M_{cos} has more extreme values than M_{cosrw}; as explained below, it has both more zero cosine values, and more very high values. This is most likely because M_{cosrw}, from doing the walk, picks up exercises that are another degree of separation away. When the row of the matrix is normalized to capture the distribution of the source node, the otherwise high values from M_{cos} are tempered in the M_{cosrw} matrix. This shift to a large number of lower scores is shown in the bottom panel of Figure 3. M_{bp} and M_{cosrw} are very similar, but M_{bp} generally has a wider dynamic range.

4.2 Comparison of the Graph Walks

Table 1 provides summary statistics for cosine similarity and the two random walks for all pairs of problems (N=3,250,809). The cosine matrix is very sparse, as shown by the median value of 0. Of the two random walk similarities, rw_{cos} has a lower standard deviation around the mean, but otherwise the two random walks produce similar distributions.

The similarity values given by cosine and the cosine random walk will increasingly differ the more that the start problem has relatively higher degree centrality due either to more words or higher frequency of words in exercises (df). For reference, the word that occurs most frequently, *number*, has a df of 0.42, and the second most frequent occurs in only 15% of the exercises. Fifty eight nodes have no edges (0 degree), the most frequent number of edges is 170, and the maximum is 1,706. Table 2 gives the summary statistics for df, number of math words, and degree centrality.

Inspection of the data shows that for pairs of problems in the two chapters for our case study, if the cosine similarity between a pair is high (≥ 0.75), the similarity values for rw_{cos} tend to go down as the number of shared word increases from 3 to between 5 and 7. For the rw_{bp}, the opposite trend occurs, where the similarity goes up as the number of words increases. This difference helps account for an observed divergence in the two graph walks for sections 5 and 6 of Chapter 6.

Table 3 illustrates two pairs of problems from section 5 that have high cosine similarities, and relatively higher rw_{bp} similarities (greater than the rw means of 0.0055) and relatively lower rw_{cos} (lower than the rw means). The reverse pattern is seen for two pairs of problems from section 6 that have high cosine similarities. These problems have higher than average rw_{cos} and lower than average rw_{bp}. What differentiates the two pairs of problems is that the section 5 problems have a relatively large number of words in common: 14 for the first pair, 12 for the second pair. In both pairs, some of the words have relatively high document frequency. As discussed above, these two properties increase the degree centrality of the start node of a step in the rw_{cos} graph, and thus lower the probability of hitting each of the start node's one-degree neighbors. This effect propagates along the two steps of the walk. For the rw_{bp} graph, however, as the number of shared math words for a pair of problems increases, the number of paths from one to the other also increases, thus raising the probability of the traversal. This effect also propagates through a two-step walk. In contrast to the section 5 problems, the two section 6 problems have relatively fewer words in common: 3 for both pairs.

For problem pairs where the cosine similarity is between 0.40 and 0.60, the mean similarity from rw_{bp} is 30% higher than for rw_{cos} for when the number of math words in common is 3 (0.0033 vs. 0.0043), 80% higher when the number of math words in common is 6 (0.0024 versus 0.0045), and three as high when the number of math words in common is 9 (0.0023 versus 0.0068). For problems pairs where the cosine similarity is less than 0.20, the two walks produce very similar results. The average similarity values for the bipartite walk are about 20% higher, and the maximum values are higher, but the two walks produce similar means, independent of the lengths of the common word vectors, or the total number of math words.

Table 2: Summary statistics of document frequency (df) of math words, number of math words in problems, and degree centrality of rw_{cos}

	df	math words	degree ctr. rw_{cos}
minimum	5.54×10^{-4}	1	0
maximum	0.424	24	1,706
mean	0.183	8.35	418.4
median	6.66×10^{-3}	8.00	340
std. dev.	0.314	3.64	317.1

Table 3: Two pairs of problems with high cosine similarity and reverse patterns of graph walk similarity. The first pair, from section 5, have lower than average rw_{cos} and higher than average rw_{bp} due to relatively many words in common (12 and 14). The second pair, from section 6, have higher than average rw_{cos} and lower than average rw_{bp} due to relatively few words in common.

Prob 1	Prob 2	cosine	rw_{cos}	rw_{bp}	N	max df
6.5.99	6.5.85	1.0000	0.0032	0.0102	12	0.42
6.5.94	6.5.83	0.8819	0.0026	0.0064	14	0.13
6.6.109	6.6.102	0.8660	0.0068	0.0037	3	0.11
6.6.104	6.6.102	0.7746	0.0068	0.0029	3	0.11

Since we normalized the matrices across rows, which are the source nodes, differences between the bipartite matrix, M_{bp}, and the cosine matrices implied that the degree of the target node had a greater impact on the variability in the bipartite matrix. To measure the impact of the edge degree on the target nodes, we considered the column sum for those targets that had 1 edge, those that had 2, etc. up to 20 edges. The results are summarized in Figure 4. As can be seen, the column sum varies linearly by the number of target edges in the bipartite matrix, whereas the cosine matrices do not. We found the cubed root of the column sum in M_{bp} approaches the distribution of column sums of the cosine matrices, which is provided in Figure 4.

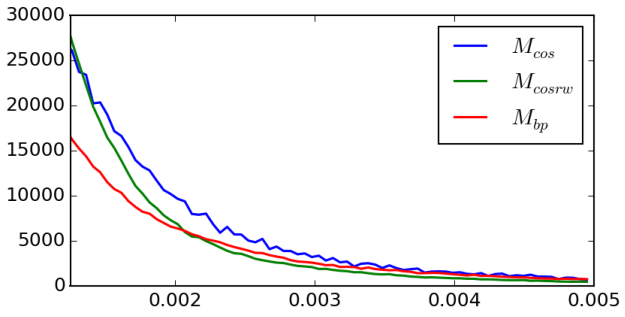

Figure 3: Tail distribution of similarity values in M_{cos}, M_{cosrw}, and M_{bp}. Because 62% of the values in M_{cos} are 0, the plot shows only non-zero values.

5. CONCLUSION

Visualization of the three similarity matrices shows they reveal the same overall patterns, thus each is confirmed by the others. However, the bipartite walk was the most sensitive to word frequency across exercises, and the number of words in problems. With our goal of automatically discovering knowledge components and identifying their relationships, the random walk that stepped in proportion to its cosine similarity performed best. It was able to discover second-degree relationships that seem reasonable as we explore by eye those matches. Future work will test these re-

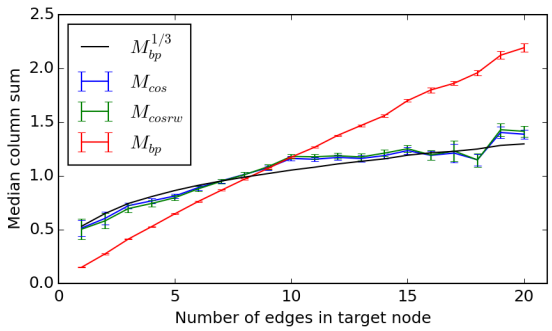

Figure 4: Distribution of column sums by number of edges in the target node represented by the column. Error plots show the mean and standard error for each type. Black line is the cubed root of the mean of the column sums of M_{bp}.

lationships with student performance data. We should find, for example, that if two exercises are conceptually similar, then student outcomes should also be similar and learning curves should reveal shared knowledge components. In this respect, such automatically constructed knowledge graphs can create more refined domain models that intelligent tutoring systems and robust assessments can be built upon.

6. REFERENCES

[1] Y. An, J. Janssen, and E. E. Milios. Characterizing and mining the citation graph of the computer science literature. *Knowledge and Information Systems*, 6(6):664–678, 2004.

[2] J. Berant, V. Srikumar, P.-C. Chen, A. Vander Linden, B. Harding, B. Huang, P. Clark, and C. D. Manning. Modeling biological processes for reading comprehension. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1499–1510, Doha, Qatar, October 2014. Association for Computational Linguistics.

[3] P. Buitelaar, A. Frank, M. Hartung, and S. Racioppa. Ontology-based information extraction and integration from heterogeneous data sources, 2008.

[4] A. Carlson, J. Betteridge, B. Kisiel, B. Settles, E. H. Jr., and T. M. Mitchell. Toward an architecture for never-ending language learning. In *Proceedings of the 24th Conference on Artificial Intelligence (AAAI)*, volume 2, pages 1306–1313, 2010.

[5] G. Erkan and D. R. Radev. Lexrank: Graph-based lexical centrality as salience in text summarization. *J. Artif. Int. Res.*, 22(1):457–479, Dec. 2004.

[6] B. Hixon, P. Clark, and H. Hajishirzi. Learning knowledge graphs for question answering through conversational dialog. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Denver, CO, May-June 2015.

[7] R. J. L. John, R. J. Passonneau, and T. S. McTavish. Semantic similarity graphs of mathematics word problems: Can terminology detection help? In *Proceedings of the Eighth International Conference on Educational Data Mining*, 2015.

[8] K. R. Koedinger, A. T. Corbett, and C. Perfetti. The knowledge-learning-instruction (KLI) framework: Toward bridging the science-practice chasm to enhance robust student learning. *Cognitive Science*, 36(5):757–798, 2012.

[9] K. Krippendorff. *Content analysis: An introduction to its methodology*. Sage Publications, Beverly Hills, CA, 1980.

[10] N. Kushman, Y. Artzi, L. Zettlemoyer, and R. Barzilay. Learning to automatically solve algebra word problems. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 271–281, Baltimore, Maryland, June 2014. Association for Computational Linguistics.

[11] N. Li, W. W. Cohen, and K. R. Koedinger. Discovering student models with a clustering algorithm using problem content. In *Proceedings of the 6th International Conference on Educational Data Mining*, 2013.

[12] E. Minkov, W. W. Cohen, and A. Y. Ng. Contextual search and name disambiguation in email using graphs. In *Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '06, pages 27–34, New York, NY, USA, 2006. ACM.

[13] I. Niles and A. Pease. Mapping WordNet to the SUMO ontology. In *Proceedings of the IEEE International Knowledge Engineering Conference*, pages 23–26, 2003.

[14] J. Otterbacher, G. Erkan, and D. R. Radev. Biased lexrank: Passage retrieval using random walks with question-based priors. *Information Processing and Management*, 45(1):42–54, 2009.

[15] L. Page, S. Brin, R. Motwani, and T. Winograd. The pagerank citation ranking: Bringing order to the web. Technical Report 1999-66, Stanford InfoLab, November 1999. Previous number = SIDL-WP-1999-0120.

[16] T. M. Pilehvar, D. Jurgens, and R. Navigli. Align, disambiguate and walk: A unified approach for measuring semantic similarity. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1341–1351. Association for Computational Linguistics, 2013.

[17] J. Scheffczyk, A. Pease, and M. Ellsworth. Linking FrameNet to the suggested upper merged ontology. In B. Hennett and C. Fellbaum, editors, *Formal Ontology in Information Systems*, pages 289–. IOS Press, 2006.

[18] M. J. Seo, H. Hajishirzi, A. Farhadi, and O. Etzioni. Diagram understanding in geometry questions. In *Proceedings of the 28th AAAI Conference on Artificial Intelligence*, 2014.

[19] M. Šimko and M. Bieliková. Automatic concept relationships discovery for an adaptive e-course. In *Proceedings of the Second International Conference on Educational Data Mining (EDM)*, pages 171–179, 2009.