Using Correlational Topic Modeling for Automated Topic Identification in Intelligent Tutoring Systems

Stefan Slater
Ryan Baker
University of Pennsylvania
3700 Walnut St.
Philadelphia, PA 19104
{slater.research, ryanshaunbaker}@gmail.com

Ma. Victoria Almeda
Alex Bowers
Teachers College Columbia University
525 W. 120th St.
New York, NY 10027
{mqa2000,bowers}@tc.columbia.edu

Neil Heffernan
Worcester Polytechnic Institute
100 Institute Rd.
Worcester, MA 01609
nth@wpi.edu

ABSTRACT
Student knowledge modeling is an important part of modern personalized learning systems, but typically relies upon valid models of the structure of the content and skill in a domain. These models are often developed through expert tagging of skills to items. However, content creators in crowdsourced personalized learning systems often lack the time (and sometimes the domain knowledge) to tag skills themselves. Fully automated approaches that rely on the covariance of correctness on items can lead to effective skill-item mappings, but the resultant mappings are often difficult to interpret. In this paper we propose an alternate approach to automatically labeling skills in a crowdsourced personalized learning system using correlated topic modeling, a natural language processing approach, to analyze the linguistic content of mathematics problems. We find a range of potentially meaningful and useful topics within the context of the ASSISTments system for mathematics problem-solving.

CCS Concepts
• **Information systems~Document topic models**

Keywords
Topic Modeling; Correlational Topic Modeling; Natural Language Processing; Mathematics Education; Intelligent Tutoring Systems

1. INTRODUCTION
Accurate estimation of student knowledge in online learning environments generally relies on the existence of skill model frameworks that map specific problems to a broader theme, topic, or skill. Models such as Bayesian Knowledge Tracing (BKT; [9]) and Performance Factors Analysis (PFA; [18]) utilize skill models as an underlying structure for drawing inferences about student knowledge. However, this process of associating problems in an intelligent tutoring system (ITS) with annotations of the skills and knowledge that are associated with the problem benefits more than just knowledge tracing models. Teachers and researchers can use information about the underlying skill model to determine if skills differ in important ways, such as determining which skills are more likely to be associated with disengagement (e.g. [1]) or negative affect [11], and to study how hint requests and other metacognitive behaviors vary between different skills [24]. Skill models can also help to inform teachers' assessments of student learning and performance, and identify areas where students may need additional practice or scaffolding in order to succeed.

However, the human creation and curation of models of domain structure can be challenging [22] especially in online learning environments which utilize crowdsourced or teacher-generated problem content. With the rising interest in scaling high-quality online education, there is also increasing effort to engage a broader community including teachers in creating content [12]. However, teachers frequently lack the time and training to produce high-quality annotations of the skills associated with specific content, and there are no guarantees that skill tags will be consistent between different authors using a crowd-sourced system. Definitions of skills, and the granularity of skills associated with particular problems, may vary from author to author, and render the overall skill model across the system uninterpretable, or worse, inconsistent.

While newer knowledge estimation approaches such as recurrent neural networks (RNNs) do not require expert-coded domain structure knowledge or skill models [19], it is unclear whether RNNs offer a tangible increase in performance for knowledge estimation compared to more traditional approaches such as Bayesian Knowledge Tracing (BKT) (see discussion in [15]). Additionally, RNNs have poor interpretability, as their predictions cannot be straightforwardly tied to specific skills or features of the problems themselves. As such, RNNs are an incomplete substitute for having a skill-problem mapping. When it is not feasible to manually author the mapping between skills and problems, automatically deriving this model may have substantial value.

There have been efforts to automate the process of determining which skills are associated with each of a set of problems by using the co-occurrence of correctness across problems. For example, [2] derived the mapping between test items and latent skills by taking several initial mappings (with randomized restart), testing their fit to the data, and using a search algorithm to enhance the mapping. Other approaches to automatically deriving mappings between problems and skills, such as those used by [10] and [23], utilize matrix factorization to infer skills based on student responses. These approaches assume that student patterns of correct and incorrect responses have less variance within skills than between skills, and that this difference in variance can be used to draw inferences about the latent skill structure of the data. This approach has been shown to have high accuracy, provided

LAK '17, March 13-17, 2017, Vancouver, BC, Canada © 2017 ACM.
ISBN 978-1-4503-4870-6/17/03☐$15.00
DOI: http://dx.doi.org/10.1145/3027385.3027438

that (1) there are relatively few latent skills in the data and (2) that the skills are substantially different from one another. These assumptions are difficult to maintain in a specialized mathematics tutor that considers potentially hundreds of individual skills, all within the same overall domain. Another limitation to these approaches is that they typically assume that knowledge is static when analyzing covariance, but knowledge in online learning environments is in fact changing as it is being measured.

An alternative to methods utilizing patterns of correct and incorrect answers is topic modeling approaches, such as Latent Semantic Analysis (LSA; [16]), Latent Dirichlet Analysis (LDA; [5]), and Correlational Topic Modeling (CTM; [4]) Topic models are not dependent on human tagging of skills – the only input required is the textual content of the problem itself. Using topic modeling approaches to label skills and topics utilizes both the relationships between words and symbols within problem texts, as well as inferred knowledge about the absence of particular words, a similar approach to how students themselves come to understand and learn material [16]. Novice learners often come to understand complex material by first learning via semantic, surface-level features, and relies heavily on information that is available perceptually [13]. Therefore, an approach which utilizes the textual information contained in a problem may serve to better map to students' emerging understanding of skills, rather than experts' higher-order determinations. Additionally, this approach can model the underlying lexical similarity that exists within problems that share a common skill.

Topic modeling is a form of natural language processing that utilizes word co-occurrence patterns to identify clusters of words, called "topics", that appear in large collections of documents. Topic modeling can be loosely characterized as factor analysis conducted on words, rather than numerical variables. Topic models have been used for a range of applications, such as automated matching of reviewers to scientific papers within particular fields of study [17] and utilizing latent topics within capital facility finance bond elections to examine which topics are more likely to pass [6].

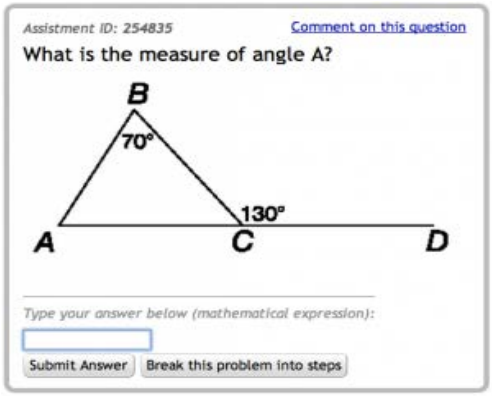
From this family of models, we select correlated topic modeling (CTM) [4], which models the intercorrelations of words in text to infer topics – as the most appropriate modeling approach to use for our data. LDA assumes that all topics are present in differing proportions across the component documents [3], an assumption which almost certainly doesn't hold across the scope of math topics present within ASSISTments. Additionally, mathematics problems tend to be highly focused on a single or very small selection of topics, and so modeling the proportionality of topics between problems is unnecessary. Finally, although LDA can model proportions of topics present in a document or documents, it cannot measure the *correlations* that may be present between documents – for example, a math problem about trigonometry is much more likely to involve geometry than fractions. LSA suffers from this same limitation. This is the aspect of CTM that makes it valuable for this particular data – because of the possible relationships existing between skills, we expect a strong underlying correlational structure between the topics that we are attempting to model. Because of this, as well as the wide scope of problems within our dataset, we selected CTM as our modeling approach of choice.

2. METHODS

Data for this research comes from the ASSISTments platform [12]. ASSISTments is an online intelligent tutoring system used by over 50,000 students, primarily in the northeastern United States. ASSISTments provides formative and summative assessment, as well as student support, scaffolding, and detailed teacher reports.

Within the ASSISTments system, students work through problem sets, consisting of mathematics questions which they provide answers to. There are several types of problem sets: complete all problem sets, which require all problems to be answered correctly; if-then-else problem sets, which require a certain percentage of problems to be answered correctly; and skill builder problem sets, which require students to answer three consecutive problems correctly in a row. Within most problems, students have the opportunity to request hints from the system. On many problems, students providing incorrect answers are asked to complete scaffolding problems – sub-problems which work on a specific aspect or skill associated with the base problem. Problem responses are generally fill in the blank, but can also be multiple choice or short answers. Problem sets can be assigned as classwork, to be completed with the supervision of a teacher, or as homework, to be completed when the student has study hall or is at home.

Figure 1. Example of a problem in ASSISTments



We analyzed the textual content of 112,526 problems, nearly all of them mathematics problems, and most of them developed and used by teachers in the 2012-2013 school year. In ASSISTments, teachers author problems and hints using a text editor, and have the ability to add in mathematical symbols, images, embed video, and use rich HTML formatting.

2.1 Text Processing

Preprocessing of text is an important step for NLP approaches. ASSISTments text data includes HTML tags and markup and HTML characters, which must be removed before they can be used in modeling. Additionally, because a majority of the problems included numbers and mathematical symbols, we had to develop additional coding methods for numbers and mathematical symbols.

CTM is a bag-of-words approach to text analysis – it solely considers what words are present, and does not consider ordering or other relationships between words, or any special qualities of the words themselves. This approach has advantages and disadvantages to working with the data for this research. For example, special symbols such as the degree symbol or the square root operator are HTML encoded (as ° and &sqrt;. A bag of words model will treat these strings like any other word, while

more sophisticated tools which tag words with semantic categories or other linguistic information (e.g. [20]), may struggle to represent this information correctly. However, there are disadvantages, especially when working with numeric information. A bag of words tagger will differentiate between 1, 2, 3, and so on, even when there is no reason to think, for example, that the equations 4 + 3 and 2 + 5 are different in any relevant way.

To address this limitation of the modeling approach, and to attempt to capture differences in meaning within the mathematics problems that would not be readily apparent to a pure bag-of-words approach, we developed several dummy codes for the data. These dummy codes were implemented using string search and replace routines. The original text, and their replacements, can be found in Table 1. We selected our converted text notation in such a way that it would not resemble actual words and phrases used in the problems, This approach was necessary in order to reduce the variance that the CTM would attempt to identify between numbers, decimals, and degrees which are semantically very similar.

Table 1. Conversions between raw texts and their corresponding labels in the data

Original Text	Converted Text
{0-9}	xxnumxx
{10+}	xxmanynumxx
Decimals, e.g. 1.11	xxdecimalxx
Fractions, e.g. 2/3	xxfracxx
Dollar Amounts, e.g. $3.50	xxmoneyamtxx
Percents, e.g. 76%	xxpercentxx
Degree Amounts, e.g. 90°	xxdegreesxx
"Explicit" Numbers, e.g. #4[1]	xxexplnumxx

[1] **Teachers often used this notation to denote questions which came from a worksheet or textbook that students had access to.**

After performing string replacement, all HTML elements within the problems – such as embedded videos, image links (URLs), and font changes – were removed. These HTML elements contained little data that could be meaningfully parsed by a bag of words model – URLs, for instance, were almost always unique by problem. Additionally, the textual content contained within the HTML tags was not visible to the learner. Punctuation and mathematical operators were all removed, as well as excess whitespace. While punctuation and mathematics operators serve a distinct purpose within the text, there was too much inconsistency between use cases and problem authors' conventions for this information to be extracted reliably. The R package topicmodels was used to remove common grammatical words and words which contained less than three characters. This package was also used to stem the dataset (converting words such as contained, contains, and containing to contain – [14]. A sparse matrix was created with d = 0.9999, removing all words which appeared in less than 0.0001% of problems. These steps produced a final dataset consisting of 4,058 words mapped to 112,526 problems.

Construction of the model was performed in R. Term frequency weightings were applied to the document term matrix, and three CTM models were calculated – one with five predicted topics, one with 15 predicted topics, and one with 25 predicted topics. Because of the exploratory nature of this work, optimal tf-idf weightings and optimal numbers of topics were not calculated. Tf-idf (term frequency/inverse document frequency) weightings can help with the identification and exclusion of extremely common or extremely uncommon words, as well as weight frequencies of common and uncommon words more effectively. The current skill label proxy variable within ASSISTments estimates close to 330 skills – a number which was not computationally feasible for our hardware at this time. In future work, we plan to continue this work on a cloud computing setup. Goodness of the resulting models was calculated via the perplexity of each model. Perplexity scores measure the ability of topic models to generalize to new and unseen text (in the case of these models, a test set of problems). A perplexity score can be thought of as the number of probable words that could follow any given word within the model, therefore, a lower perplexity represents a better model fit.

3. RESULTS

The perplexity scores for each of the three models are presented in Table 2. The 25-topic model was found to have the lowest perplexity among the three models tested, indicating that is it the best fit among the three models presented here. The downward trend in perplexity for higher K suggests that additional topics could more appropriately model problem content within ASSISTments.

Table 2. Perplexity scores for the three topic models

K =	Perplexity
5	319.91
15	227.40
25	189.28

The 25 topics identified, along with the five most common words associated with each topic, are presented in Table 3.

Table 3. 25 Topics identified by algorithm

Topic	Correlated Terms	Topic Label
1	Many, student, xxpercentxx, look, take	Number/percentage conversion[1]
2	Left, attempt, xxexplnumxx, xxmanynumxx, xxdecimalxx	System generated – you have XX attempts left.
3	Origin, problem, let, try, solution	System generated, after scaffolding – "Let's try the original problem"
4	Xxdecimalxx, express, divide, paper, point	Teacher reminder – "express your answer as a decimal to the hundredths point"
5	Step, one, problem, break, button	System reminder – "do not press break this problem into steps"
6	Question, sorry, next, incorrect, attempt	System generated – "Sorry, that's incorrect. Let's move onto the next question"
7	Fraction, number, answer, mix, improper	Improper and mixed fractions
8	Triangle, angle, length, figure, side	Side length and angles of triangles
9	Xxexplnumxx, page, unit, xxmanynumxx	Textbook and worksheet problems; "Page 25 #4"
10	Equation, line, variable, write, slope	Slope problems
11	Nearest, round, place, answer, hundredth	Teacher reminder – "round your answer to

		the nearest hundredth"
12	Best, choose, follow, part, two	A vs. B comparison problems
13	Day, xxnumxx, time, play, month	Time problems
14	Xxmanynumxx, point, score, game, name	Sports problems
15	Xxmoneyamtxx, number, cost, answer, total	Currency questions
16	Xxnumxx, power, xxdecimalxx, xxmanynumxx, number	Metric explanation problems[2]
17	Answer, make, type, fraction, enter	Teacher reminder – how to enter fraction answers
18	Area, xxnumxx, scale, square, xxdecimalxx	Area problems
19	Xxfracxx, number, whole, fraction, example	Whole fraction problems
20	Mile, xxnumxx, per, ball, car	Distance problems
21	Xxnumxx, divid, conversion, formula, number	Unit conversion problems
22	Xxnumxx, xxmanynumxx, number, find, value	Simple algebra problems
23	Xxdecimalxx, fraction, numerator, multiply, denominator	Decimal – fraction conversion problems
24	Xxnumxx, factor, simplify, follow	Factorization problems
25	Follow, correct, select, subtract, label	Instructions about subsequent parts of a problem

[1] **These problems often took the form of interpreting a pie chart, and calculating the number of students who constituted a given percentage of the overall population.**

[2] **These problems scaffolded students by explaining the nature of the metric system in base ten, using decimals to show that relationship. "Powers of ten" was a common phrase as well.**

There are a number of surprising findings in the CTM results. First, we expected that CTM would distinguish between different topics within problems, which it appears to be able to do. For example, topics 10 (slope problems), 15 (currency problems), 21 (unit conversion problems), and 23 (fraction conversion problems) appear to be well-formed. What we didn't anticipate, however, was that CTM would *also* pick up on common phrases or hints that the system provided to students, such as topic 2 (reminders about the number of attempts a student has left), topics 4 and 11 (reminders about significant figures), and topic 3 (when a student returns to an original problem after a scaffolding problem).

Additionally, the model appears to identify non-mathematics themes, such as topic 14 (sports) and topic 15 (currency). Similar approaches to identifying the semantic content of problems have been used before [21], and topic modeling may be an additional approach to identifying themes that are present within problems. While these categories and the reminders/hints to students don't lend themselves towards the goal of automated skill tagging of problems, they are interesting for assessing features of problem construction, such as the use of reminder texts and feedback about student performance, or the use of specific themes in word problems. If CTM is able to reliably tag these features, then it is possible to use them in models assessing the relationship between problem design and student affect, learning, and behavior. As such, even the categories which are less useful for our initial research goal are likely to have other potentially productive uses.

4. CONCLUSION

In this paper we have developed a correlational text model (CTM) to attempt to identify common topics within a mathematics tutor. These approaches are important for being able to estimate student knowledge, as well as for guiding and informing teacher feedback and identification of student performance gaps. We developed a CTM which used 25 topics, and determined that it had better model fit than 15- and 5-topic alternatives. The CTM was able to identify not only mathematics subjects such as fraction problems, slope problems, and area problems, but also instances of system-generated scaffolding and hints (such as reminders about rounding) and non-mathematics subjects, such as problems concerning sports and money.

This approach is not without its limitations though – one of our goals in this effort was to develop an automated method of skills tagging, and the results of the CTM are somewhat murky in that respect. Only 9 of the 25 topics identified appear to be about a clearly defined mathematics skill/concept, the rest of the topics identifying either system-generated text or non-mathematics subjects. In other words, the CTM attempts to identify differences in problem content, problem structure and problem theme, all at the same time, and the resulting model is somewhat muddy as a result. It is unclear whether an increase in the number of topics will capture more skills and find more fine-grained skills, or just add more noise and variance to the underlying model.

It is also likely that this is not an optimal mapping of the domain structure of these problems. Performing CTM beyond 25 topics was computationally limiting, but the ASSISTments system identifies roughly 330 unique skills within the database. Future efforts at using CTM to identify domain structure will need to utilize cluster or cloud computing, as this approach is computationally demanding. However, this expansion of topics comes with a cost – research by Chang et al. suggests that, while CTM tends to outperform LSA and LDA in terms of model fit and word intrusion metrics, it does so at the cost of human interpretability. In other words, while CTM does a better job of clustering topics than LSA and LDA, the topics themselves may not be as interpretable to human judges, especially as the number of topics to be modeled increases [7].

An additional limitation of this modeling approach is the depth of structure that can be assessed. As the CTM does not take into account the order of words, grammar, and rhetoric, it may provide an oversimplified categorization of the math problems within the tutoring system. In particular, this approach may combine problems with similar surface features, but with different deep structure - the underlying principle that is necessary for a solution [8]. Analysis and tagging of deep structure of problems, therefore, currently still needs to be more reliant on human coders and expert judgment than the automated approach used here; trying to expand the depth of our categorizations with more sophisticated linguistic approaches will be an important area of future work.

Finally, mathematics notation represents something of an unsolved problem. While string replacement worked to a degree, improvement in the identification of topics in mathematics problems will benefit from an improvement in the capacity for text analysis tools to work with and process mathematical notation, such as equations, unit notations (such as ft, in, km), and variables (none of which were captured with the current text replacement scheme). However, the analysis of mathematics text

is not well-developed and this work further highlights the necessity for specialized tools for exploring data involving mathematics symbols and notation with NLP tools. Differences in the uses of numbers, equations, variables, operators, and symbols represent a large source of potential variance and structure within the data that cannot currently be explored in an effective way, and an enhanced ability to parse mathematics items in text could greatly enhance the ability of topic models to successfully distinguish individual skills.

Future work in this domain may attempt to use the results from a CTM as a skill matrix for various knowledge inference techniques, such as BKT or PFA. If CTM and other forms of topic modeling can achieve an acceptable level of agreement with expert skill tagging, then we should expect to see improved model fit for these NLP-derived skill models compared to previous methods for automated skill tagging, at lower cost and time to implement than manual skill tagging. These improvements in the scalability of skill tagging would serve to improve the quality and consistency of skill identification in ITS environments, improving both the quality of personalized learning while making it easier for researchers to develop models that build on skill models and use these models to understand and enhance student learning.

5. ACKNOWLEDGMENTS

This research was supported by the National Science Foundation (NSFDRL 1252297). Any opinions and findings expressed are the authors' and do not necessarily reflect the views of the NSF. ASSISTments has benefited from multiple US DOE and NSF grants and awards. Our thanks to the University of Pennsylvania Research Apprenticeship Course, for their feedback and thoughts on earlier drafts of this paper. Our thanks also to Vitomir Kovanovic for his insights on dealing with mathematics equations and texts within the data.

6. REFERENCES

[1] Baker, R.S.J.d., de Carvalho, A.M.J.A., Raspat, J., Aleven, V., Corbett, A.T., Koedinger, K.R. (2009) Educational Software Features that Encourage and Discourage "Gaming the System". *Proceedings of the 14th International Conference on Artificial Intelligence in Education*, 475-482.

[2] Barnes, T., Bitzer, D., & Vouk, M. (2005). Experimental analysis of the q-matrix method in knowledge discovery. In *International Symposium on Methodologies for Intelligent Systems*. Springer Berlin Heidelberg, 603-611.

[3] Blei, D. M. (2012). Probabilistic topic models. *Communications of the ACM*, *55*(4), 77-84.

[4] Blei, D. M., & Lafferty, J. D. (2007). A correlated topic model of science. *The Annals of Applied Statistics*, 17-35.

[5] Blei, D. M., Ng, A. Y., & Jordan, M. I. (2003). Latent dirichlet allocation. *Journal of Machine Learning Research*, *3*(Jan), 993-1022.

[6] Bowers, A.J., Chen, J.(2015) Ask and Ye Shall Receive? Automated Text Mining of Michigan Capital Facility Finance Bond Election Proposals to Identify which Topics are Associated with Bond Passage and Voter Turnout. *Journal of Education Finance*, 41(2), 164-196.

[7] Chang, J., Gerrish, S., Wang, C., Boyd-Graber, J. L., & Blei, D. M. (2009). Reading tea leaves: How humans interpret topic models. In *Advances in neural information processing systems*, 288-296.

[8] Chi, M. T., Glaser, R., & Farr, M. J. (2014). *The nature of expertise*. Psychology Press, xvii-xxi.

[9] Corbett, A. T., Anderson, J. R. (1995). Knowledge tracing: Modeling the acquisition of procedural knowledge. *User Modeling and User-Adapted Interaction 4*, 4, 253–278.

[10] Desmarais, M. C. (2012). Mapping question items to skills with non-negative matrix factorization. *ACM SIGKDD Explorations Newsletter*, *13*(2), 30-36.

[11] Doddannara, L., Gowda, S., Baker, R.S.J.d., Gowda, S., de Carvalho, A.M.J.B (2013) Exploring the relationships between design, students' affective states, and disengaged behaviors within an ITS. *Proc. of the 16th International Conference on Artificial Intelligence and Education*, 31-40.

[12] Heffernan, N. T., & Heffernan, C. L. (2014). The ASSISTments ecosystem: building a platform that brings scientists and teachers together for minimally invasive research on human learning and teaching. *Int'l. Journal of Artificial Intelligence in Education*, *24*(4), 470-497.

[13] Hmelo-Silver, C. E., & Pfeffer, M. G. (2004). Comparing expert and novice understanding of a complex system from the perspective of structures, behaviors, and functions. *Cognitive Science*, *28*(1), 127-138.

[14] Jivani, A. G. (2011). A comparative study of stemming algorithms. *Int. J. Comp. Tech. Appl*, *2*(6), 1930-1938.

[15] Khajah, M., Lindsey, R. V., & Mozer, M. C. (2016). How deep is knowledge tracing? *Proc. of the 9th International Conference on Educational Data Mining,* 94-101.

[16] Landauer, T. K., Foltz, P. W., & Laham, D. (1998). An introduction to latent semantic analysis. *Discourse processes*, *25*(2-3), 259-284.

[17] Mimno, D., & McCallum, A. (2007). Expertise modeling for matching papers with reviewers. *Proc. 13th ACM SIGKDD Conf. on Knowledge Discovery and Data Mining,* 500-509.

[18] Pavlik Jr, P. I., Cen, H., & Koedinger, K. R. (2009). Performance Factors Analysis--A New Alternative to Knowledge Tracing. *Proceedings of AIED 2009,* 531-538.

[19] Piech, C., Bassen, J., Huang, J., Ganguli, S., Sahami, M., Guibas, L. J., & Sohl-Dickstein, J. (2015). Deep knowledge tracing. In *Advances in Neural Inf. Processing Sys.* 505-513.

[20] Rayson, P. (2008). Wmatrix corpus analysis and comparison tool. Lancaster University.

[21] Slater, S., Ocumpaugh, J., Baker, R., Scupelli, P., Inventado, P.S., Heffernan, N. (2016) Semantic Features of Math Problems: Relationships to Student Learning and Engagement. *Proceedings of the 9th International Conference on Educational Data Mining.*, 223-230.

[22] Stamper, J. C., & Koedinger, K. R. (2011). Human-machine student model discovery and improvement using DataShop. *International Conference on Artificial Intelligence in Education*. Springer Berlin Heidelberg, 353-360.

[23] Thai-Nghe, N., Horváth, T., & Schmidt-Thieme, L. (2010). Factorization models for forecasting student performance. *Proceedings of Educational Data Mining 2011*, 11-20.

[24] Walkington, C., Clinton, V., Ritter, S. N., & Nathan, M. J. (2015). How readability and topic incidence relate to performance on mathematics story problems in computer-based curricula. *J. of Educational Psychology*,*107*(4).