Learning Representations for Math Strategies using BERT

Abisha Thapa Magar
University of Memphis
Memphis, TN, USA
thpmagar@memphis.edu

Stephen E. Fancsali
Carnegie Learning
Pittsburgh, PA, USA
sfancsali@carnegielearning.com

Vasile Rus
University of Memphis
Memphis, TN, USA
vrus@memphis.edu

April Murphy
Carnegie Learning
Pittsburgh, PA, USA
amurphy@carnegielearning.com

Steve Ritter
Carnegie Learning
Pittsburgh, PA, USA
sritter@carnegielearning.com

Deepak Venugopal
University of Memphis
Memphis, TN, USA
dvngopal@memphis.edu

ABSTRACT

Adapting to a student's problem solving strategy can lead to improved engagement and motivation. In this work, we develop an AI-based approach to analyze math learning strategies at scale. Specifically, we use a state-of-the-art AI model, namely, BERT to learn structure within strategies observed in large datasets. In particular, we consider the MATHia ITS and define strategies as sequences of steps that a student follows in solving the problem. We apply BERT pre-training to learn semantic representations of strategies from a workspace in MATHia that allows for different strategies. Further, we fine-tune these embeddings to train them on downstream tasks such as identifying a strategy and understanding drift in strategy. Our preliminary results are encouraging and demonstrate that BERT can uncover hidden structure in strategies and therefore is a promising direction to analyze large-scale math learning data.

CCS CONCEPTS

• **Computing methodologies** → **Artificial intelligence**; • **Applied computing** → **Interactive learning environments**.

KEYWORDS

learning representations; BERT; Math learning strategies; big data

ACM Reference Format:
Abisha Thapa Magar, Stephen E. Fancsali, Vasile Rus, April Murphy, Steve Ritter, and Deepak Venugopal. 2024. Learning Representations for Math Strategies using BERT. In *Proceedings of the Eleventh ACM Conference on Learning @ Scale (L@S '24), July 18–20, 2024, Atlanta, GA, USA.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3657604.3664711

1 INTRODUCTION

Intelligent Tutoring Systems (ITSs) [23] and Adaptive Instructional Systems (AISs) aim to provide personalized instruction to large and diverse student populations. In contrast to traditional classrooms, individual instruction can be tailored to learners, closely monitor their performance and provide timely feedback. ITSs can help scale-up individual instruction making high-quality education more accessible to all. There is evidence that such one-on-one instruction can be more effective than traditional classrooms where students may have varying levels of skill [23]. To improve the effectiveness of personalized math instruction in ITSs, a fundamental research task is to understand how students solve math problems, i.e., the approach that students take while thinking through a problem. The problem-solving approach of a student is a highly complex function dependent on many factors such as experience with similar problems, general expertise in the topic, other cognitive abilities, social factors, etc. Adapting to a specific learner by understanding their problem solving approach accounts for differences in learning abilities, learning styles and education goals. Such adaptation often leads to more engaging and effective learning [30]. To this end, human experts have explored factors that influence problem-solving strategies used by students [16]. However, human experts are expensive and limited in the ability to analyze large data from thousands, tens of thousands, or millions of students. Advanced AI methods and access to large computing infrastructure such as the cloud offer new possibilities to analyze in-depth and at scale large learner datasets to help us analyze and discover new insights into Math problem solving strategies.

AI has revolutionized several domains over the last decade. For instance, *Imagenet* [13], *embeddings* [15] and generative AI [19] are some of the key transformative developments made possible by a combination of large-scale data along with powerful models. Inspired by these, in this paper, we present our initial findings on learning Bi-Directional Encoding Representations of Transformers (BERT) [8] models on data collected from MATHia, a well-known ITS for Math learning. Specifically, we define strategies based on sequences of steps that students perform while interacting with MATHia. We then use BERT to learn representations for strategies through *step masking*. Specifically, we *pre-train* the BERT model to predict masked steps to learn an embedding over strategies. We present initial results for a 7th grade workspace in MATHia with data from around 3K students where we have *ground-truth* that multiple strategies exist based on the workspace design. Our results demonstrate that BERT embeddings can uncover structure within the strategies. Further, we also fine-tune the pre-trained model to identify strategies and also to analyze how strategies shift over time. Our results demonstrate that our approach is a promising direction to scale up analysis of math strategies over large datasets.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
L@S '24, July 18–20, 2024, Atlanta, GA, USA
© 2024 Copyright held by the owner/author(s). Publication rights licensed to ACM.
ACM ISBN 979-8-4007-0633-2/24/07
https://doi.org/10.1145/3657604.3664711

2 BACKGROUND

Ritter et al. [21] provide a comprehensive survey on different approaches used to identify student strategies. Classical model tracing [4] based methods can be used to identify if certain strategies can improve learner mastery. In [20], based on model tracing, an approach was proposed that can evaluate the effectiveness of an unknown strategy. Based on its effectiveness, the strategy can be used to augment the set of known correct/incorrect strategies and thus incrementally build the set of labeled strategies. Several discriminative learning methods have been explored to identify strategies. For instance, labeled datasets were created based on students interacting with the ITS [5] and in each of these cases, the interactions are labeled manually based on one/more experts. Such labeled datasets can be used for training Machine Learning models. However, the quality of strategy identification is heavily dependent upon the feature specification [18]. In [6], a Machine learning model was learned from data labeled from Cognitive Tutor and similar approaches have been used in several other contexts. For instance, in [7], a Machine learning model was used for inferring strategies from SQL-tutor, in [9] for strategies in role playing games, in [22] for strategies in understanding conceptual physics, etc. Sequence modeling has often been used to detect strategies from sequential datasets. Specifically, Betty's brain [1] is a virtual environment where students learn about scientific processes by teaching a virtual agent called Betty. This is an open-ended learning environment where students have the flexibility to perform their tasks in a variety of different ways which yields different strategies. Other studies have used this environment to identify strategies through sequence mining [10, 11]. In [31], sequence pattern mining was applied to a MOOCs platform to analyze activity sequences of learners [31]. Shakya et al. [25, 26] developed a scalable approach to train LSTMs using interaction data from Mathia. In conversational agents, we can view strategies in the space of dialogue acts [3] and/or actions [2]. Therefore, one approach that has been explored is to map language into sequences of dialogue acts and identify higher-level pedagogical *modes* from these sequences [14, 24, 29]. These modes give a high level overview of pedagogical intentions behind acts.

3 APPROACH

In the context of ITSs, the strategies followed by a learner are highly dependent on the design choices made in the ITS. For instance, an ITS could restrict its design such that everyone follows the same strategy. If the design offers choices to a student, different learners could emerge with different strategies. To analyze strategies, we first need to define them more precisely. In a top-down approach, we can define strategies at an abstract level (e.g. guess and check, pattern finding, tabular methods, etc.) and then ground these strategies in a specific context to gain insights such as how do learners use a specific strategy, which ones are more beneficial, etc. In a data-driven approach, given observations, we could extract strategies from data and explain them using domain knowledge. The AI based methods are more naturally suited to the latter. Specifically, in the context of MATHia, learners perform actions, where each action represents a learner's progress towards completing a *goal node* or *step* in the solution. Strategies in this context are *sequences of steps* that the learner completes based on performing actions within the action-space [17].

Fig. 1 (a) shows a problem from the workspace in 7th Grade math for calculating *percent increase and percent decrease*. The learner completes each of the slots which are considered as steps, each of which is associated with a *skill* (knowledge component [12]). Further, as shown in Fig. 1 (b), the learner can perform *optional tasks*. These tasks offer additional scaffolding to help the learner. Given this design, we specify a *ground truth* over *types* of strategies as follows. i) Students could perform the optional steps and therefore learn strategies as prescribed by the optional steps to solve the problem. For instance, in the context of the above example, they would most likely utilize the equivalent ratios method, the means and extremes method or both to solve the problem. ii) On the other hand, they could bypass the scaffolding and use their prior knowledge or a non-standard approach to complete their solution.

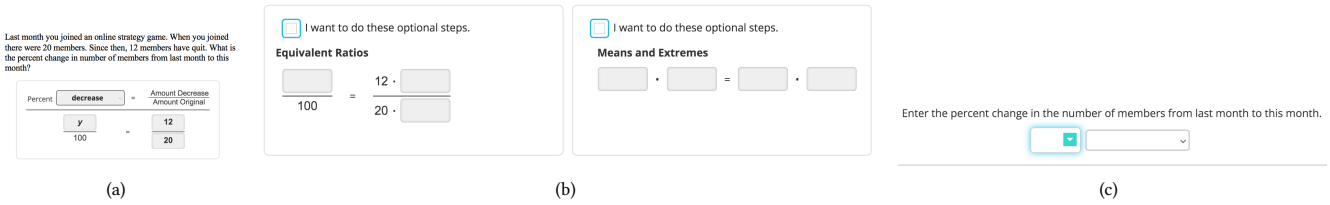
A simple approach to represent a strategy is to represent it as a *one-hot* encoded vector ($\mathbf{V}$) of the sequence of steps performed by a learner. However, this representation is not semantically meaningful. Specifically, students can perform steps in different orders, repeat the steps, etc. Therefore, we want to learn a representation where sequences that are semantically similar have similar representations. To learn a *dense* representation (or *embedding*) for strategies, we can use a hypothesis similar to the *distributional hypothesis* in language learning, i.e., for a pair of steps say s_1, s_2, if the context in which s_1 is used is consistently similar to the context in which s_2 is used over large datasets, then we can assign similar embeddings to s_1, s_2. Here, context of a step s refers to other steps in the strategy that are performed before or after s in the strategy. The dimensionality of an embedding is typically much smaller compared to the dimensionality of $\mathbf{V}$.

3.1 BERT Representation

We learn BERT models using logs collected from MATHia that records student actions. There are two key steps in the BERT model. First, we perform *pre-training* and then *fine-tuning* of the model. The pre-training phase uses unlabeled data and the goal is to infer structure within the data while the fine-tuning is used to perform a specific task and is typically supervised with a small amount of labeled data.

Pre-Training. The original BERT model uses a *Masked Language Model* (MLM) pre-training objective to infer language structure. Similarly, here, we perform pre-training where we mask steps in the strategy and train using an objective that infers strategy structure. Specifically, each step in the strategy is treated as a token (after performing simple pre-processing such as removing repeated steps that indicate multiple student tries at the same step, removing steps that are not directly related to problem solving, etc.). We mask tokens in the steps and predict masked tokens using the BERT model. The BERT model learns a hidden representation for each of the masked tokens and then predicts the masked token through a softmax distribution over the vocabulary of tokens.

The model underlying BERT is a bidirectional transformer [28] which uses context in both directions. Specifically, it uses context information for a masked step from preceding as well as succeeding



(a) (b) (c)

Figure 1: An example problem from the workspace on computing percent increase/decrease. (a) shows the problem and the steps (each step is a fillable slot that the learner completes), (b) shows optional tasks that provide additional scaffolding using equivalent ratios or means and extremes and (c) shows steps in the final solution.

steps and therefore learns a more semantically accurate representation of the steps compared to models that are uni-directional. For instance, generative models such as OpenAI's GPT [19] generates tokens one at a time, where each token is conditioned on the prior tokens and therefore are unidirectional. Further, the BERT model uses *positional encodings* which help us encode invariance in strategies. For instance, in Fig. 1 (a), actions where the numerator is filled in before the denominator is invariant with actions where the denominator is filled in before the numerator. Positional encodings help us learn a representation that is invariant to re-orderings that are not significantly different in terms of meaning. In our case, this meaning is automatically inferred by the model by learning from a large number of strategies. The BERT model combines the token embeddings for steps in the strategies with the positional encoding that represents the approximate positions of these steps to learn the final embedding for the strategy.

Fine-tuning. We fine-tune a pre-trained model to perform specific prediction tasks. For the fine-tuned models, we add a layer on top of the pre-trained model to predict the task-specific output and train the model end-to-end on the task-specific input, output pairs. In our case, we used two types of fine-tuning, namely i) strategy type identification and ii) predicting strategy drift. In strategy type identification, we train the model using cross-entropy loss.

$$\ell(\theta) = -\sum_{i=1}^{N} y_i \log(P_\theta(\mathbf{X}_i) + (1 - y_i) \log(1 - P_\theta(\mathbf{X}_i)) \quad (1)$$

where N is the number of instances used for fine-tuning, $y_i = 1$ if the student uses an optional-task based strategy and 0 otherwise, θ are the model parameters, $\mathbf{X}_i$ is a sequence of tokens describing learner actions.

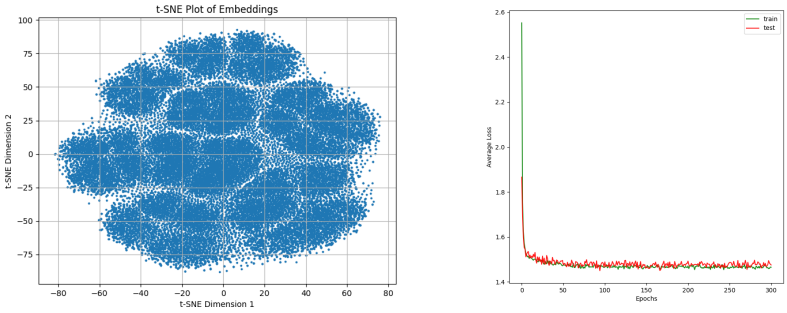
Similarly, we also fine-tune the model to predict drift in strategies, i.e., we want to know if strategies change over time. To do this, we use the same loss function as in Eq. (1). However, in this case, we fine-tune two constrained models by sampling strategies either from i) the start of the problem-solving session (IS) or ii) towards the end of the problem-solving session (FS) in the workspace. In the IS model, we want identify if initial strategies can predict final strategies and in the FS model we want identify if final strategies can predict initial strategies.

3.2 Experiments

We demonstrate some preliminary results of our approach on the 7-th Grade workspace related to percentage increase or decrease. The workspace has around 300 problems and 3K students completed this workspace. The data is available through the PSLC datashop [27]. There were around 45K unique instances (student, problem pairs) in this workspace. Each recorded interaction consists of the log of the student's action toward solving the problem. For example, the timestamp, step name, the knowledge component, the number and type of hints used and if the step was completed correctly along with several other pieces of information. We pre-processed this dataset to remove repeated occurrences of the same step, i.e., if a step indicates a bug/error the student may repeat it several times, however, all of these actions represent a single step in the strategy. Further, we also removed auto-completed entries since these are not part of a strategy. That is, in several cases MATHia auto-completes certain steps when the student completes a problem and we do not want these to be part of the student's strategy.

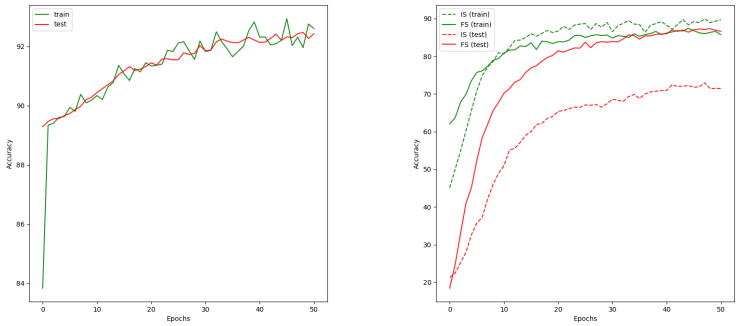
Pre-training. We used the publicly available BERT implementation in our experiments. We used 8 self-attention heads and 4 encoder blocks. We used a 64 dimension embedding (as compared to 512 in the original model) since the vocabulary is much more limited which significantly reduced training time (#parameters $\approx$ 200K). We randomly masked 15% of tokens to pre-train the BERT models (similar to what has typically been used in BERT). We stopped pre-training when the validation loss was below a specified threshold. We used a Tesla GPU for training the model on the Google Vertex AI platform. The time taken for the pre-training to converge was around 4.5 hours. Fig. 2 (a) shows the TSNE plot for the embeddings (the embeddings projected into 2D) after the pre-training converged. As seen in the figure, the data shows separation of strategy embeddings which is evidence that there is underlying strategy structure in the workspace. Further, the training and validation curves in Fig. 2 (b) illustrate that the pre-training converges which illustrates that the BERT model can predict masked tokens accurately and again, indicates that there is a predictable structure within strategies.

Fine-tuning. We show the results of the two tasks that fine-tune the BERT embeddings. In each case, we fine-tune the model by supervising the pre-trained model with labeled data. Note that the pre-trained model can be fine-tuned very fast since it already has knowledge of strategies. In particular, we took just around 15 minutes for fine-tuned training in each task. Fig. 3(a) shows our results for identifying strategy types. As shown here, on unseen test data, we could identify the strategy type with high accuracy (the peak accuracy was around 92% where the test set was a balanced



(a) (b)

Figure 2: (a) TSNE plot for pre-trained strategy embeddings (b)Training and validation loss in pre-training.



(a) (b)

Figure 3: Fine-tuned model performance. (a) Identifying strategy types (b) Predicting strategy drift. IS indicates training initial strategy to predict final strategy and FS indicates training final strategy to predict initial strategy.

set). The 5-fold cross-validated F1 score was 0.87. Fig. 3 (b) shows our results in predicting strategy drift. Here, we sample the initial 10% of problems that a student has worked on to train the IS model and predict the results on the final 10% of problems within the workspace. Similarly, in FS, we train on the final 10% of problems that a student has worked on and test on the initial 10% of problems. As seen in the figure, FS is more accurate than IS. The F1-score for FS was 0.93 and for IS it was 0.83. This indicates that the strategy drifts but at the same time learners retain preferences of their original strategy (may indicate prior experience or knowledge). We plan to interpret strategy drift more deeply in follow up work.

4 CONCLUSION AND FUTURE WORK

We presented an approach to analyze math strategies at scale using BERT. Specifically, we considered the MATHia ITS and defined strategy as a sequence of steps performed by a student. Based on this definition, we considered a specific workspace in 7th Grade math and identified types of strategies that students could follow on this workspace. We pre-trained a BERT model by masking steps in a strategy to understand the general structure of strategies and then fine-tuned it across 2 tasks, strategy identification and analyzing strategy drift. We demonstrated results using data from MATHia for around 3K students. Our results showed the following, i) there exists structure within strategies which can be discovered by BERT and ii) while we can recognize strategies with fairly high accuracy, it may be harder to predict the drift of strategies.

Based on promising preliminary results, we will perform experiments covering several workspaces in MATHia. Further, we will also develop interpretations to connect well known strategies (top-down methods) with observations from our approach. Finally, we will connect strategies with learning outcomes.

ACKNOWLEDGMENTS

This research was supported by an award from the Bill and Melinda Gates Foundation and NSF award #2008812. The opinions, findings, and results are solely the authors' and do not reflect those of the funding agencies.

REFERENCES

[1] Gautam Biswas, James R. Segedy, and Kritya Bunchongchit. 2016. From Design to Implementation to Practice a Learning by Teaching System: Betty's Brain. *International Journal of Artificial Intelligence in Education* 26, 1 (2016), 350–364.

[2] Kristy Elizabeth Boyer, Eun Young Ha, Robert Phillips, Michael D. Wallis, Mladen A. Vouk, and James C. Lester. 2010. Dialogue act modeling in a complex task-oriented domain. In *Proceedings of the 11th Annual Meeting of the Special Interest Group on Discourse and Dialogue*. 297–305.

[3] Whitney L. Cade, Jessica L. Copeland, Natalie K. Person, and Sidney K. D'Mello. 2008. Dialogue Modes in Expert Tutoring. In *Proceedings of the 9th International Conference on Intelligent Tutoring Systems (ITS)*. 470–479.

[4] Albert T. Corbett. 2001. Cognitive Computer Tutors: Solving the Two-Sigma Problem. In *Proceedings of the 8th International Conference on User Modeling 2001*. 137–147.

[5] Ryan Shaun Joazeiro de Baker, Albert T. Corbett, and Angela Wagner. 2006. Human Classification of Low-Fidelity Replays of Student Actions. In *Proceedings of the Educational Data Mining Workshop at the 8th International Conference on Intelligent Tutoring Systems*.

[6] Ryan Shaun Joazeiro de Baker and Adriana M. J. B. de Carvalho. 2008. Labeling Student Behavior Faster and More Precisely with Text Replays. In *Educational Data Mining 2008, The 1st International Conference on Educational Data Mining, Montreal, Québec, Canada, June 20-21, 2008. Proceedings*, Ryan Shaun Joazeiro de Baker, Tiffany Barnes, and Joseph E. Beck (Eds.). www.educationaldatamining.org, 38–47.

[7] Ryan Shaun Joazeiro de Baker, Antonija Mitrovic, and Moffat Mathews. 2010. Detecting Gaming the System in Constraint-Based Tutors. In *User Modeling, Adaptation, and Personalization, 18th International Conference, UMAP 2010, Big Island, HI, USA, June 20-24, 2010. Proceedings (Lecture Notes in Computer Science, Vol. 6075)*, Paul De Bra, Alfred Kobsa, and David N. Chin (Eds.). Springer, 267–278.

[8] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, Jill Burstein, Christy Doran, and Thamar Solorio (Eds.). Association for Computational Linguistics, 4171–4186.

[9] Kristen E. DiCerbo and Khusro Kidwai. 2013. Detecting Player Goals from Game Log Files. In *Proceedings of the 6th International Conference on Educational Data Mining, Memphis, Tennessee, USA, July 6-9, 2013*, Sidney K. D'Mello, Rafael A. Calvo, and Andrew Olney (Eds.). International Educational Data Mining Society, 314–315.

[10] John S. Kinnebrew and Gautam Biswas. 2012. Identifying Learning Behaviors by Contextualizing Differential Sequence Mining with Action Features and Performance Evolution. In *Proceedings of the 5th International Conference on Educational Data Mining, Chania, Greece, June 19-21, 2012*, Kalina Yacef, Osmar R. Zaïane, Arnon Hershkovitz, Michael Yudelson, and John C. Stamper (Eds.). www.educationaldatamining.org, 57–64.

[11] John S. Kinnebrew, Kirk M. Loretz, and Gautam Biswas. 2013. A Contextualized, Differential Sequence Mining Method to Derive Students' Learning Behavior Patterns. *Journal of Educational Data Mining* 5, 1 (2013), 190–219.

[12] Kenneth R. Koedinger, Albert T. Corbett, and Charles Perfetti. 2012. The Knowledge-Learning-Instruction Framework: Bridging the Science-Practice Chasm to Enhance Robust Student Learning. *Cogn. Sci.* 36 (2012), 757–798.

[13] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. 2012. ImageNet Classification with Deep Convolutional Neural Networks. In *Advances in Neural Information Processing Systems*, F. Pereira, C.J. Burges, L. Bottou, and K.Q. Weinberger (Eds.), Vol. 25. Curran Associates, Inc.

[14] Nabin Maharjan, Dipesh Gautam, and Vasile Rus. 2018. Assessing Free Student Answers in Tutorial Dialogues Using LSTM Models. In *Artificial Intelligence in Education - 19th International Conference, AIED*. 193–198.

[15] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed Representations of Words and Phrases and their Compositionality. In *Neural Information Processing Systems*. 3111–3119.

[16] Gyöngyvér Molnár, Samuel Greiff, and Benő Csapó. 2013. Inductive reasoning, domain specific and complex problem solving: Relations and development. *Thinking Skills and Creativity* 9 (2013), 35–45.

[17] Allen Newell and Herbert Simon. 1972. *Human Problem Solving*. Prentice Hall.

[18] Luc Paquette, Adriana M. J. B. de Carvalho, Ryan S. Baker, and Jaclyn Ocumpaugh. 2014. Reengineering the Feature Distillation Process: A case study in detection of Gaming the System. In *Proceedings of the 7th International Conference on Educational Data Mining, EDM 2014, London, UK, July 4-7, 2014*, John C. Stamper, Zachary A. Pardos, Manolis Mavrikis, and Bruce M. McLaren (Eds.). International Educational Data Mining Society (IEDMS), 284–287.

[19] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language Models are Unsupervised Multitask Learners. (2019). https://openai.com/blog/better-language-models/

[20] Steve Ritter. 1997. Communication, cooperation and competition among multiple tutor agents. In *Artificial Intelligence in Education: Knowledge and media in learning systems*. 31–38.

[21] Steve Ritter, Ryan Baker, Vasile Rus, and Gautam Biswas. 2019. Identifying Strategies in Student Problem Solving. *Design Recommendations for Intelligent Tutoring Systems* 7 (2019), 59–70.

[22] Elizabeth Rowe, Ryan S. Baker, Jodi Asbell-Clarke, Emily Kasman, and William J. Hawkins. 2014. Building Automated Detectors of Gameplay Strategies to Measure Implicit Science Learning. In *Proceedings of the 7th International Conference on Educational Data Mining, EDM 2014, London, UK, July 4-7, 2014*, John C. Stamper, Zachary A. Pardos, Manolis Mavrikis, and Bruce M. McLaren (Eds.). International Educational Data Mining Society (IEDMS), 337–338.

[23] Vasile Rus, Sidney K. D'Mello, Xiangen Hu, and Arthur C. Graesser. 2013. Recent Advances in Conversational Intelligent Tutoring Systems. *AI Magazine* 34, 3 (2013), 42–54.

[24] Vasile Rus, Nabin Maharjan, Lasang Jimba Tamang, Michael Yudelson, Susan R. Berman, Stephen E. Fancsali, and Steven Ritter. 2017. An Analysis of Human Tutors' Actions in Tutorial Dialogues. In *Proceedings of the International Florida Artificial Intelligence Research Society Conference*. 122–127.

[25] Anup Shakya, Vasile Rus, and Deepak Venugopal. 2021. Student Strategy Prediction using a Neuro-Symbolic Approach. In *Proceedings of the 14th International Educational Data Mining Conference (EDM 21)*.

[26] Anup Shakya, Vasile Rus, and Deepak Venugopal. 2023. Scalable and Equitable Math Problem Solving Strategy Prediction in Big Educational Data. In *Proceedings of the 16th International Educational Data Mining Conference (EDM 23)*.

[27] John C. Stamper, Kenneth R. Koedinger, Ryan Shaun Joazeiro de Baker, Alida Skogsholm, Brett Leber, Sandy Demi, Shawnwen Yu, and Duncan Spencer. 2011. DataShop: A Data Repository and Analysis Service for the Learning Science Community. In *AIED*, Vol. 6738. 628.

[28] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is All you Need. In *NIPS*. 5998–6008.

[29] Deepak Venugopal and Vasile Rus. 2016. Joint Inference for Mode Identification in Tutorial Dialogues. In *COLING 2016, 26th International Conference on Computational Linguistics*. ACL, 2000–2011.

[30] Shuai Wang, Claire Christensen, Wei Cui, Richard Tong, Louise Yarnall, Linda Shear, and Mingyu Feng. 2020. When adaptive learning is effective learning: comparison of an adaptive learning system to teacher-led instruction. *Interactive Learning Environments* 0, 0 (2020), 1–11.

[31] J. Wong, Mohammad Khalil, M. Baars, B. D. Koning, and F. Paas. 2019. Exploring sequences of learner activities in relation to self-regulated learning in a massive open online course. *Computers & Education* (2019).