MathBERT:A Pre-trained Language Model for General NLP Tasks in Mathematics Education

Jia Tracy Shen
Penn State University
jqs5443@psu.edu

Michiharu Yamashita
Penn State University
michiharu@psu.edu

Ethan Prihar
Worcester Polytechnic Institute
ebprihar@wpi.edu

Neil Heffernan
ASSISTments.org
Worcester Polytechnic Institute
neil@ASSISTments.org

Xintao Wu
University of Arkansas
xintaowu@uark.edu

Ben Graff
Stride, Inc
bgraff@k12.com

Dongwon Lee
Penn State University
dongwon@psu.edu

Abstract

Since the introduction of the original BERT (i.e., BASE BERT), researchers have developed various customized BERT models with improved performance for specific domains and tasks by exploiting the benefits of *transfer learning*. Due to the nature of mathematical texts, which often use domain specific vocabulary along with equations and math symbols, we posit that the development of a new BERT model for mathematics would be useful for many mathematical downstream tasks. In this paper, we introduce our multi-institutional effort (i.e., two learning platforms and three academic institutions in the US) toward this need: MathBERT, a model created by pre-training the BASE BERT model on a large mathematical corpus ranging from pre-kindergarten (pre-k), to high-school, to college graduate level mathematical content. In addition, we select three general NLP tasks that are often used in mathematics education: prediction of knowledge component, auto-grading open-ended Q&A, and knowledge tracing, to demonstrate the superiority of MathBERT over BASE BERT. Our experiments show that MathBERT outperforms prior best methods by 1.2-22% and BASE BERT by 2-8% on these tasks. In addition, we build a mathematics specific vocabulary 'mathVocab' to train with MathBERT. We release MathBERT for public usage at: https://github.com/tbs17/MathBERT.

1 Introduction

The arrival of transformer-based language model, BERT [4], has revolutionized the NLP research and applications. One strength of BERT (denoted as BASE BERT) is its ability to adapt to a new domain and/or task through pre-training by means of so-called "transfer learning." By taking advantage of this benefit, therefore, researchers have adapted BERT into diverse domains (e.g., FinBERT [15], ClinicalBERT [9], BioBERT [11], SCIBERT [1], E-BERT [24], LiBERT [6]) and tasks (e.g., [22], [21], [2], [14], [7]) with improved performances. In the domain of mathematics, as mathematical text often use domain or context specific words, together with math equations and symbols, we posit that mathematics-customized BERT would help researchers and practitioners sort out the meaning of ambiguous language better by using surrounding text to establish "math" context. Further, such an

35th Conference on Neural Information Processing Systems (NeurIPS 2021), Workshop on Math AI for Education (MATHAI4ED)

improved context-aware understanding of language could help develop and improve solutions for challenging NLP tasks in mathematics.

For example, some of the challenging tasks are: (i) large-scale knowledge component (KC, a.k.a. skill) prediction (denoted as T_{kc}), (ii) open-ended question answer scoring (i.e., auto-grading) (denoted as T_{ag}), and (iii) knowledge tracing (KT) correctness prediction (denoted as T_{kt}). The struggle with T_{kc} is partly attributed to its tediousness and labor-intensive work to label all knowledge components in problem texts. The traditional way to address this challenge is to use machine learning to classify them via feature extraction [10, 17, 18], which has produced decent results. For open-ended question answer grading tasks, although not favored by educators due to its difficulty of developing universal automated grading support, it is still important to develop an effective solution. Similarly, *Knowledge Tracing*, a very important task in the education domain, is defined as the task of tracing students' knowledge state to predict students' next question correctness based on their past learning activities. The current solutions [3, 23, 13, 12, 16] tend to rely on high-dimensional sequential data but not able to capture the complex nature of students' learning activities over extended periods of time.

Addressing this lack of general BERT-based language model in mathematics education, we introduce **MathBERT**, a model created by pre-training the BASE BERT model on a large mathematical corpus ranging from pre-kindergarten (pre-k), to high-school, to college graduate level mathematical content. We hypothesize that a BERT model trained on mathematical domain corpora could be more effective in mathematics-related tasks than the BASE BERT. That is, we further *pre-train* the BASE BERT on mathematical corpora to build MathBERT. Then, we use the pre-trained weights from MathBERT to *fine-tune* on the mathematical task-specific text dataset for classification. We make the following contributions in this work:

1. We build MathBERT by pre-training the BASE BERT on mathematical domain texts ranging from pre-k to high-school to graduate level mathematical curriculum, books and paper abstracts.
2. We create and release a custom vocabulary `mathVocab` to reflect the different nature of mathematical corpora (e.g., mathematical equations and symbols) and examine its effects on the three general NLP tasks.
3. We evaluate the performance of MathBERT via T_{kc}, T_{ag} and T_{kt}, and compare its performance to three groups of baseline models. MathBERT outperforms prior best models by 1.18-21.99% and BASE BERT by 2.01-8.28%.

2 Building MathBERT

BERT model pre-trained on a domain corpus is called DAPT (Doman-adaptive Pre-training) BERT and the ones pre-trained on a task specific corpus is called TAPT (Task-adaptive Pre-training) BERT [7]. The difference between TAPT and DAPT BERT training is in the size of the input data (see illustration in Appendix B Fig. 2). DAPT models have much larger corpora whereas TAPT models are more specific to tasks.

Math Corpora: MathBERT is a DAPT model that is pre-trained on a large mathematics related corpora comprising mathematics curricula from pre-k to high school, mathematics textbooks written for high school and college students, mathematics course syllabi from Massive Online Open Courses (MOOC) as well as mathematics paper abstracts (see in Table 1). We crawl these data from popular mathematics curriculum websites (illustrativemathematics.org, utahmiddleschoolmath.org, engageny.org), a free text book website (openculture.com), a MOOC platform (classcentral.com), and arXiv.org, with a total data size of around 3GB and 100 million tokens. The mathematics corpora not only contain text but also mathematics symbols and equations. Among these data, the text book data is in PDF format and we hence convert them into text format using the Python package `pdfminer`[1], which preserves the mathematics symbols and equations to an extent (see corpus sample in Appendix A Fig. 1)

Further pre-training: To further train MathBERT efficiently, we adopt a similar data processing strategy to the ROBERTa model, which threaded all the sentences together and split them into a maximum length of 512-token sequence sections [14]. Inspired by SciBERT [1], we create a

[1] https://pypi.org/project/pdfminer/

Table 1: Math Corpus Details. Note all the corpus is in mathematics domain

Source	Math Corpora	Tokens
arxiv.org	Paper abstract	64M
classcentral.com	College MOOC syllabus	111K
openculture.com	pre-k to College Textbook	11M
engageny.org	Pre-k to HS Curriculum	18M
illustrativemathematics.org	K-12 Curriculum	4M
utahmiddleschoolmath.org	G6-8 Curriculum	2M
ck12.org	K-12 Curriculum	910K

custom mathematical vocabulary (`mathVocab`) using Hugging Face `BertWordPieceTokenizer`[2] with a token size of 30,522 same as the size of original vocabulary file (denoted as `origVocab`) from BASE BERT [4]. `mathVocab` is richer in the representation of mathematics equation and symbols than the `origVocab` (see vocabulary difference in Appendix A Table 4). We use 8-core TPU machine from Google Colab Pro to further train the BASE BERT with batch size (bs) of 128, the learning rate (lr) of $5e-5$, and maximum sequence length (max-seq) of 512 for MathBERT with `origVocab` (MathBERT-orig) and MathBERT with `mathVocab` (MathBERT-custom). The training effectiveness is measured via Mask Language Modeling (MLM) accuracy (ACC) of 99.8%, where the model predicts the vocabulary ID of the masked words in a sentence [4]. It takes 5 days and 600K steps to finish training for each model. Both MathBERT-orig and MathBERT-custom model artifacts are released in Tensorflow and Pytorch versions (see download details in `https://github.com/tbs17/MathBERT`).

3 Downstream Math NLP Tasks

We use three mathematical tasks mentioned in Section 1 to demonstrate the usefulness of MathBERT. (i) KC Prediction (T_{kc}): a single sentence *multinominal classification* problem (213 labels) with $Input(I) \mapsto text$ and $Output(O) \mapsto KC$ (i.e., one of 213 labels). (ii) Auto-grading (T_{ag}): a two-sentence *multinominal classification* problem (5 labels) with $I \mapsto Question\&Answer$ pair and $O \mapsto Score$. (iii) KT Correctness (T_{kt}): a two-sentence *binary classification* problem with $I \mapsto Question\&Answer$ pair and $O \mapsto Correctness$.

Data: The data for T_{kc}(denoted as D_{kc}) has 13,722 math problem texts with their correspondent skill codes (213 labels). The data for T_{ag}(denoted as D_{ag}) has 141,186 open-ended math problem texts with the correspondent scores (1-5) and the data for T_{kt}(denoted as D_{kt}) has 269,230 math problem texts with the correspondent correctness (correct/incorrect) (see the text and label examples in Appendix A Table 5). All these data are provided by ASSISTments [8] and match the data sets in the best performing prior work [21, 5, 12] for each task. In particular, the KT data is the text version (269,230 texts and 2 labels) of the ASSISTments 2009 data[3], the numeric form of which was used by the best performing prior work [12].

Further training and Fine-tuning : We adopt TAPT training strategy to further pre-train the task-data, using only the text part and achieving above 99% MLM accuracy (see in Appendix B Table 7). For fine-tuning, both texts and labels are used in a split ratio of 72% training, 8% validating, and 20% testing to predict the labels (see Table 2). We apply all three data sets onto BASE BERT, TAPT BERT, MathBERT-orig and MathBERT-custom models individually and optimize hyper-parameter tuning (see in Appendix B Table 6) to achieve their best results.

4 MathBERT Evaluation

To evaluate the effectiveness of MathBERT (both MathBERT-orig and MathBERT-custom) across three tasks, we fine-tune MathBERT on the three tasks and compare its performance to baseline models (Prior, BASE BERT, TAPT in Table 3). F1, ACC (i.e., Accuracy) and AUC are used to

[2]https://huggingface.co/docs/tokenizers/python/latest/quicktour.html

[3]https://sites.google.com/site/assistmentsdata/home/assistment-2009-2010-data/skill-builder-data-2009-2010

Table 2: Task Data Details. KC: Knowledge Component, KT: Knowledge Tracing. All data from ASSISTments platform [8]

Task	#Labels	#Texts	#Fine-tune Split		
			Train (72%)	Validate (8%)	Test (20%)
D_{kc}	213	13,722	9,879	1,098	2,745
D_{ag}	5	141,186	101,653	11,295	28,238
D_{kt}	2	269,230	193,845	21,539	53,846

Table 3: Performance Comparison: MathBERT vs. Baseline Methods across Five Random Seeds. Bold font indicates best performance and underlined values are the second best. Δ is the relative improvement of MathBERT from other baseline models: * indicates statistical significance.

Method	Vocab	T_{kc} (%)		T_{ag} (%)	T_{kt} (%)	
		F1	ACC	AUC	AUC	ACC
$PriorBest^p$	/	88.69[21]	92.51[21]	85.00[5]	81.82[12]	77.11[12]
$BASE^b$	orig	90.14	91.78	88.67	88.90	86.88
TAPT	orig	91.77	92.96	90.34	95.88	93.49
MathBERT	orig (o)	**92.67**	**93.79**	**90.57**	**96.04**	**94.07**
(m)	math (c)	<u>92.51</u>	<u>93.60</u>	<u>90.45</u>	<u>95.95</u>	<u>94.01</u>
Δ_{m-p}	orig	+4.49%	+1.38%	+6.55%	+17.38%	+21.99%
	math	+4.31%	+1.18%	+6.41%	+17.27%	+21.92%
Δ_{m-b}	orig	+2.81%***	+2.19%***	+2.14%***	+8.03%***	+8.28%***
	math	+2.63%***	+1.98%***	+2.01%***	+7.93%**	+8.21%***
$\Delta_{m^c-m^o}$	/	-0.17%	-0.20%	-0.13%	-0.09%	-0.06%

measure task prediction results so that they are consistent with the performance evaluation in the prior works for each task [10, 17, 18, 20, 12, 25, 16, 19].

After achieving the best performance for each task via hyper-parameter tuning (see detail in Appendix B Table 6), we run each model with five random seeds and report the average value which are further tested by T-test for significance. Note that we don't run five random seeds on the prior models due to the lack of accessible codes.

In Table 3, we note that MathBERT outperforms all other baseline models with MathBERT-orig achieving the best performance followed by MathBERT-custom. Particularly, MathBERT-orig has a relative improvement of 1.38% to 21.99% from the best prior methods across all metrics and tasks whereas MathBERT-custom has about 1.18% to 21.92% improvement from the prior best (see row Δ_{m-p}) . We think MathBERT's superiority performance is possibly due to the fact that MathBERT can represent the semantic and contextual features of the mathematics problem texts well via its multi-head attention heads, where it learns a feature from each head and pays attention to the important information that points to the relationship with the predicted labels. In addition, MathBERT-orig outperforms BASE BERT by about relatively 2.14 % to 8.28% with statistical significance and MathBERT-custom outperforms BASE BERT by relatively about 1.98% to 8.21% across metrics and tasks with significance (see row Δ_{m-b}). The relative improvement from BASE BERT could be due to the adaptiveness to the math domain. We don't claim that MathBERT-orig has better performance than MathBERT-custom because there's no statistical significance on the over-performance (see row $\Delta_{m^c-m^o}$).

5 Conclusion

In this work, we built and introduced MathBERT-orig and MathBERT-custom to effectively fine-tune on three mathematics-related tasks (i.e. skill code prediction, auto-grading and knowledge tracing next sentence correctness). We showed that MathBERT not only out-performed prior best methods by relatively [1.18%, 22.01%], but also proportionally out-performed the BASE BERT by [1.98%, 8.28%] with statistical significance. A mathematical vocabulary (`mathVocab`) was created and pre-trained on to reflect the special nature of mathematical corpora, which has the similar superiority over the prior model and BASE BERT performance.

6 Acknowledgement

The work was mainly supported by NSF awards (1940236, 1940076, 1940093). In addition, the work of Neil Heffernan was in part supported by NSF awards (1917808, 1931523, 1917713, 1903304, 1822830, 1759229), IES (R305A170137, R305A170243, R305A180401, R305A180401), EIR(U411B190024) and ONR (N00014-18-1-2768) and Schmidt Futures.

References

[1] Iz Beltagy, Kyle Lo, and Arman Cohan. SCIBERT: A pretrained language model for scientific text. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing and 9th International Joint Conference on Natural Language Processing*, pages 3615–3620, 2019.

[2] Minjin Choi, Sunkyung Lee, Eunseong Choi, Heesoo Park, Junhyuk Lee, Dongwon Lee, and Jongwuk Lee. MelBERT : Metaphor Detection via Contextualized Late Interaction using Metaphorical Identification Theories. In *Proceedings of the Annual Conference of the North American Chapter of the Association for Computational Linguistics*, 2021.

[3] Albert T Corbetr and John R Anderson. Knowledge Tracing: Modeling the Acquisition of Procedural Knowledge. *User Modeling and User-Adapted Interaction*, 4:253–278, 1995.

[4] Jacob Devlin, Ming Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *The Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, volume 1, pages 4171–4186, 2019.

[5] John A Erickson, Anthony F Botelho, Steven Mcateer, Ashvini Varatharaj, and Neil T Heffernan. The Automated Grading of Student Open Responses in Mathematics ACM Reference Format. In *Proceedings of the 10th Learning Analytics and Knowledge Conference*, 2020.

[6] Weiwei Guo, Xiaowei Liu, Sida Wang, Huiji Gao, Ananth Sankar, Zimeng Yang, Qi Guo, Liang Zhang, Bo Long, Bee-Chung Chen, and Deepak Agarwal. DeText: A Deep Text Ranking Framework with BERT. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, 2020.

[7] Suchin Gururangan, Ana Marasovi´cmarasovi´c, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, Noah A Smith, and Allen. Don't Stop Pretraining: Adapt Language Models to Domains and Tasks. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 2020.

[8] Neil T. Heffernan and Cristina Lindquist Heffernan. The ASSISTments ecosystem: Building a platform that brings scientists and teachers together for minimally invasive research on human learning and teaching. *International Journal of Artificial Intelligence in Education*, 24(4):470–497, 2014.

[9] Kexin Huang and Jaan Altosaar. ClinicalBert: Modeling Clinical Notes and Predicting Hospital Readmission. In *arXiv preprint arXiv:1904.05342v2*.

[10] Mario Karlovčec, Mariheida Córdova-Sánchez, and Zachary A. Pardos. Knowledge component suggestion for untagged content in an intelligent tutoring system. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 7315 LNCS:195–200, 2012.

[11] Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. Data and text mining BioBERT: a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics*, page 1234–1240, 2020.

[12] Youngnam Lee, Youngduck Choi, Junghyun Cho, Alexander R Fabbri, Hyunbin Loh, Chanyou Hwang, Yongku Lee, Sang-Wook Kim, and Dragomir Radev. Creating A Neural Pedagogical Agent by Jointly Learning to Review and Assess. In *arXiv preprint arXiv:1906.10910v2*, 2019.

[13] Qi Liu, Zhenya Huang, Yu Yin, Enhong Chen, Hui Xiong, Yu Su, and Guoping Hu. EKT: Exercise-aware knowledge tracing for student performance prediction. *IEEE Transactions on Knowledge and Data Engineering*, 33(1):100–115, 2019.

[14] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, Veselin Stoyanov, and Paul G Allen. RoBERTa: A Robustly Optimized BERT Pretraining Approach. In *arXiv preprint arXiv:1907.11692v1*, 2019.

[15] Zhuang Liu, Degen Huang, Kaiyu Huang, Zhuang Li, and Jun Zhao. FinBERT: A Pre-trained Financial Language Representation Model for Financial Text Mining. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence Special Track on AI in FinTech*, 2020.

[16] Shalini Pandey and George Karypis. A Self-Attentive model for Knowledge Tracing. In *Proceedings of The 12th International Conference on Educational Data Mining*, 2019.

[17] Zachary A Pardos. Imputing KCs with Representations of Problem Content and Context. In *Proceedings of the 25th Conference on User Modeling, Adaptation and Personalization*, pages 148–155, 2017.

[18] Thanaporn Patikorn, David Deisadze, Leo Grande, Ziyang Yu, and Neil Heffernan. Generalizability of methods for imputing mathematical skills needed to solve problems from texts. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 11625 LNAI:396–405, 2019.

[19] Chris Piech, Jonathan Spencer, Jonathan Huang, Surya Ganguli, Mehran Sahami, Leonidas Guibas, Jascha Sohl-Dickstein, Stanford University, and Khan Academy. Deep Knowledge Tracing. In *Advances in Neural Information Processing Systems*, 2015.

[20] Carolyn Rosé, Pinar Donmez, Gahgene Gweon, Andrea Knight, Brian Junker, William Cohen, Kenneth Koedinger, and Neil Heffernan. Automatic and Semi-Automatic Skill Coding With a View Towards Supporting On-Line Assessment. In *Proceedings of the conference on Artificial Intelligence in Education*, pages 571–578, 2005.

[21] Jia Tracy Shen, Michiharu Yamashita, Ethan Prihar, Neil Heffernan, Xintao Wu, Sean Mcgrew, and Dongwon Lee. Classifying Math Knowledge Components via Task-Adaptive Pre-Trained BERT. In *Proceedings of the Conference on Artificial Intelligence in Education*, 2021.

[22] Chi Sun, Xipeng Qiu, Yige Xu, and Xuanjing Huang. How to Fine-Tune BERT for Text Classification? *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 11856 LNAI(2):194–206, 2019.

[23] Nguyen Thai-Nghe, Lucas Drumond, Artus Krohn-Grimberghe, and Lars Schmidt-Thieme. Recommender system for predicting student performance. *Procedia Computer Science*, 1(2):2811–2819, 2010.

[24] Denghui Zhang, Zixuan Yuan, Yanchi Liu, Zuohui Fu, Fuzhen Zhuang, Pengyang Wang, Haifeng Chen, and Hui Xiong. E-BERT: A Phrase and Product Knowledge Enhanced Language Model for E-commerce. In *arXiv preprint arXiv:2009.02835v2*, 2020.

[25] Jiani Zhang, Xingjian Shi, Irwin King, and Dit-Yan Yeung. Dynamic Key-Value Memory Networks for Knowledge Tracing. In *International World Wide Web Conference Committee (IW3C2)*, 2017.

A Appendix

Further pre-training data sample: As mathematical corpus has a great percentage of symbols and equations, we try to preserve them as much as we can when converting from pdf format to plain text. Fig. 1 is an example of the original mathematical content before conversion.

Vocabulary Comparison: We select 50 words from the same rank tier of #2100 to #2150 and discover that `mathVocab` has more mathematical jargon than the original vocabulary (`origVocab`) from BERT [4] (see in Table 4).

1.4 Continuous Functions

We define continuous functions and discuss a few of their basic properties. The class of continuous functions will play a central role later.

Definition 1.14. *Let f be a function and c a point in its domain. The function is said to be continuous at c if for all $\epsilon > 0$ there exists a $\delta > 0$, such that $|f(c) - f(x)| < \epsilon$ whenever x belongs to the domain of f and $|x - c| < \delta$. A function f is continuous if it is continuous at all points in its domain.*

(a) Content of a Math Book

SURFACE DEFECTS IN GAUGE THEORY AND KZ EQUATION

NIKITA NEKRASOV AND ALEXANDER TSYMBALIUK

ABSTRACT. We study the regular surface defect in the Ω-deformed four dimensional supersymmetric gauge theory with gauge group $SU(N)$ with $2N$ hypermultiplets in fundamental representation. We prove its vacuum expectation value obeys the Knizhnik-Zamolodchikov equation for the 4-point conformal block of the $\widehat{\mathfrak{sl}}_N$-current algebra, originally introduced in the context of two dimensional conformal field theory. The level and the vertex operators are determined by the parameters of the Ω-background and the masses of the hypermultiplets, the cross-ratio of the 4 points is determined by the complexified gauge coupling. We clarify that in a somewhat subtle way the branching rule is parametrized by the Coulomb moduli. This is an example of the BPS/CFT relation.

(b) Abstract of a Math arXiv Paper

6.RP.A.3c

Focus Standard:	6.RP.A.3	Use ratio and rate reasoning to solve real-world and mathematical problems, e.g., by reasoning about tables of equivalent ratios, tape diagrams, double number line diagrams, or equations.
		c. Find a percent of a quantity as a rate per 100 (e.g., 30% of a quantity means 30/100 times the quantity); solve problems involving finding the whole, given a part and the percent.
Instructional Days:	6	
Lesson 24:	Percent and Rates per 100 (P)[1]	
Lesson 25:	A Fraction as a Percent (P)	
Lesson 26:	Percent of a Quantity (P)	
Lessons 27–29:	Solving Percent Problems (P, P, E)	

(c) Snippet of a Math Curriculum

Figure 1: Sample mathematical corpora text from math book, arXiv paper abstract, and curriculum

Fine-tuning data sample: The text and label examples of fine-tuning data sets D_{kc},D_{ag} and D_{kt} can be seen in Table 5. For example, the label in D_{kc} has multiple parts with '8' representing grade, 'EE' representing 'Expressions and Equations', and 'A.1' represents the lesson code.

B Appendix

TAPT vs. DAPT: According to Gururangan et al. [7], there are two styles of further pre-training on the BASE BERT [4]: (i) further pre-train the BASE BERT on a task-specific data set with tasks being text classification, question and answering inference, paraphrasing, etc. Gururangan et al. [7] call this kind of model a Task-adaptive Pre-trained (**TAPT**) Model. (ii) further pre-train the BASE BERT on a domain-specific data set with domains being finance, bio-science, clinical fields, etc. Gururangan et

Table 4: Vocabulary Comparison: origVocab vs. mathVocab. Tokens in blue are mathematics domain specific.

Vocab Type	50 Selected Tokens (from #2100-#2150)
origVocab	##y, later, ##t, city, under, around, did, such, being, used, state, people, part, know, against, your, many, second, university, both, national,##er, these, don, known, off, way, until, re, how, even, get, head, ..., didn, ##ly, team, american, because, de, ##l, born, united, film, since, still, long, work, south, us
mathVocab	cod, exist, ##olds, coun, ##lud, ##ments, squ, ##ings, known, ele, ##ks, fe, minutes, continu, ##line, addi, small, ##ology, triang, ##velop, ##etry, log, converg, asym, ##ero, norm, ##abl, ##ern, every, ##otic, ##istic, cir, ##gy, positive, hyper, dep, ##raw, ##ange, analy, equival, ##ynam, call, mon, numerical, fam, conject, large, ques, ##sible, surf

:

Table 5: Example texts of the three tasks with labels

Task Data	Label	Text
D_{kc}	8.EE.A.1	Simplify the expression: (z2)2 Put parentheses around the power if next to coefficient, for example: 3x2=3(x^2),x5=x^5
D_{ag}	5	Q: Explain your answer on the box below. A: because it is the same shape, just larger, making it similar
D_{kt}	1	Q: What is 2.6 + (-10.9)? A: -8.3

al. [7] call this kind of model a Domain-adaptive Pre-trained (**DAPT**) Model. Both TAPT and DAPT BERT models start the further pre-training process from the BASE BERT weights but pre-train on different types of corpora. TAPT BERT models pre-train on task-specific data, whereas DAPT BERT models pre-train on the domain-specific data before they are fine-tuned for use in any downstream tasks (see the process illustrated in Fig. 2).

Hyper-parameter Tuning: To acquire the best performance for the fine-tuning tasks, we search the hyper-parameter space in lr, bs, max-seq as well as epochs for each task. We discover that hyper-parameter tuning has more to do with the task data instead of the model itself. In other words, the best hyper-parameter combinations are the same across MathBERT and TAPT but vary from task to task. Table 6 shows the optimal combinations of all the hyper-parameters for each task. This result is obtained after hyper-parameter search on lr $\in \{1e-5, 2e-5, 5e-5, 8e-5, 1e-4\}$, bs $\in \{8, 16, 32, 64, 128\}$, max-seq $\in \{128, 256, 512\}$, and ep $\in \{5, 10, 15, 25\}$.

MLM Accuracy: We pre-train three TAPT models with `origVocab` from the BASE BERT [4]. Among them, $TAPT_{kc}$ and $TAPT_{ag}$ reach the best results at 100K steps and $TAPT_{kt}$ reaches its best result at 120K steps with the MLM accuracy of above 99%. We try to keep the MLM accuracy of TAPT Models similar to MathBERT (see in Table 7).

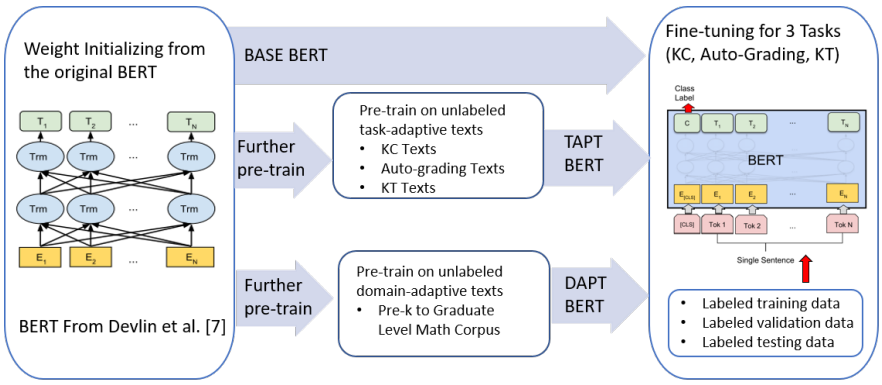


Figure 2: An illustration of training and fine-tuning process of BASE vs. TAPT vs. DAPT BERT models. The pre-training data are from this study. KC, Auto-grading, and KT Texts are task data for T_{kc}, T_{ag}, and T_{kc} respectively.

Table 6: Optimal Hyper-parameter Combination for Task fine-tuning

Task	learning rate	batch size	max sequence length	epochs
T_{kc}	5e-5	64	512	25
T_{ag}	2e-5	64	512	5
T_{kt}	5e-5	128	512	5

C Appendix

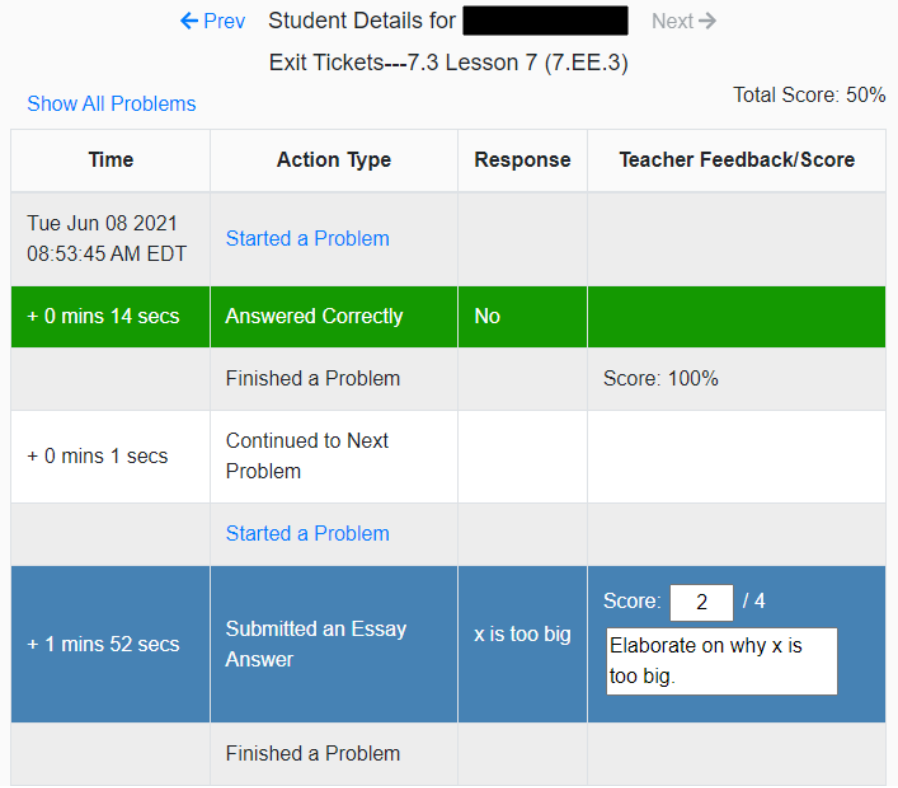
Currently, MathBERT is being adopted by two major learning management systems (i.e., ASSISTments and K12.com) to build automatic-scoring solutions to benefit teachers and students (see more in Appendix C).

ASSISTments Use Case: ASSISTments provides reports where teachers see a timeline of how each student progressed through the assignment and can grade students' open ended responses as well as leaving comments. Figure 3 shows an example of an open ended response within a student's report, together with the score and comment left by the teacher. The scores and comments will be automatically suggested by MathBERT to save time and efforts given MathBERT's state of art performance in auto-grading.

Stride Use Case: MathBERT will be used in Stride's automatic scoring pipeline where they can fine-tune MathBERT on their huge proprietary reservoir of open-ended responses and teacher feedback to automatically suggest scores and generate constructive feedback/comments for teachers to use (see more details in Fig. 4). The MathBERT output will be the suggested scores present in the unit test result (see in Fig. 5).

Table 7: Training Steps and Accuracy: MathBERT vs. TAPT vs. MathBERT+TAPT

Model	Task	Steps	MLM ACC (%)	
			origVocab	mathVocab
MathBERT	/	600K	99.85	99.95
TAPT	T_{kc}	100K	100	/
	T_{ag}	100K	99.10	/
	T_{kt}	120K	99.04	/



← Prev Student Details for [redacted] Next →

Exit Tickets---7.3 Lesson 7 (7.EE.3)

Show All Problems

Total Score: 50%

Time	Action Type	Response	Teacher Feedback/Score
Tue Jun 08 2021 08:53:45 AM EDT	Started a Problem		
+ 0 mins 14 secs	Answered Correctly	No	
	Finished a Problem		Score: 100%
+ 0 mins 1 secs	Continued to Next Problem		
	Started a Problem		
+ 1 mins 52 secs	Submitted an Essay Answer	x is too big	Score: 2 / 4 Elaborate on why x is too big.
	Finished a Problem		

Figure 3: An open response in a student's report with the teacher's score and comment.

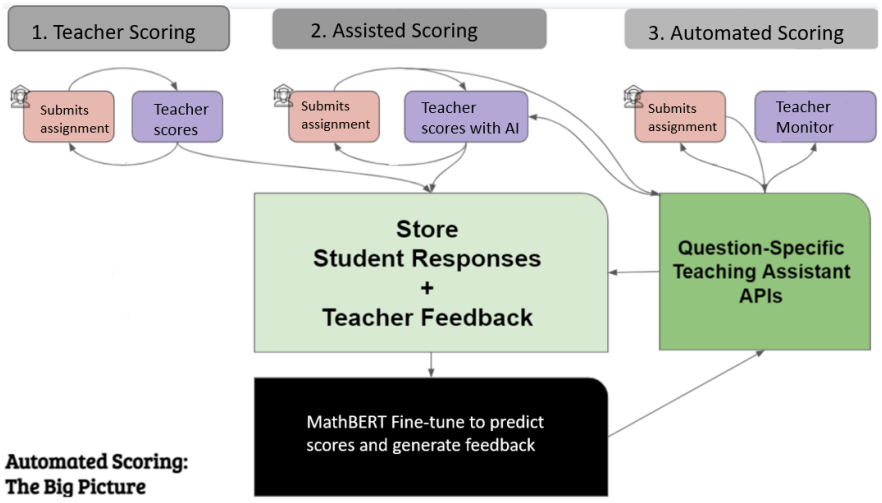
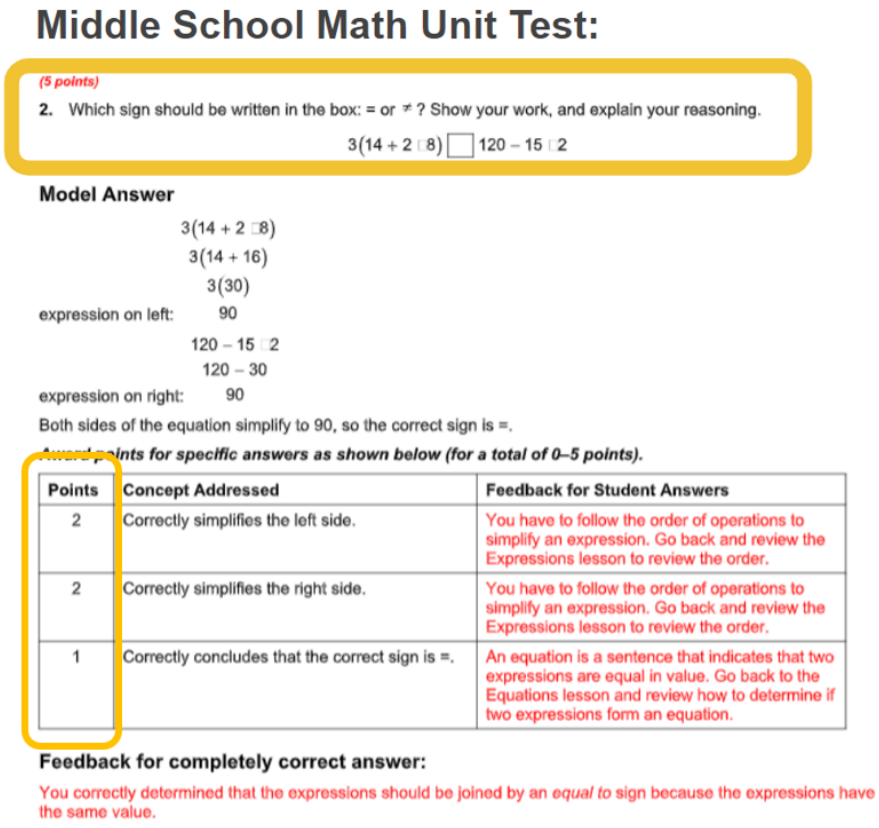


Figure 4: Stride auto-scoring pipeline using MathBERT



Middle School Math Unit Test:

(5 points)

2. Which sign should be written in the box: = or ≠ ? Show your work, and explain your reasoning.

$3(14 + 2 \square 8) \square 120 - 15 \square 2$

Model Answer

$3(14 + 2 \square 8)$
$3(14 + 16)$
$3(30)$
expression on left: 90

$120 - 15 \square 2$
$120 - 30$
expression on right: 90

Both sides of the equation simplify to 90, so the correct sign is =.

Award points for specific answers as shown below (for a total of 0–5 points).

Points	Concept Addressed	Feedback for Student Answers
2	Correctly simplifies the left side.	You have to follow the order of operations to simplify an expression. Go back and review the Expressions lesson to review the order.
2	Correctly simplifies the right side.	You have to follow the order of operations to simplify an expression. Go back and review the Expressions lesson to review the order.
1	Correctly concludes that the correct sign is =.	An equation is a sentence that indicates that two expressions are equal in value. Go back to the Equations lesson and review how to determine if two expressions form an equation.

Feedback for completely correct answer:

You correctly determined that the expressions should be joined by an *equal to* sign because the expressions have the same value.

Figure 5: Stride auto-scoring model output in the unit test