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

Jia Tracy Shen
Stride, Inc
USA
tshen@k12.com

Michiharu Yamashita
Penn State University
USA
michiharu@psu.edu

Ethan Prihar
Worcester Polytechnic Institute
USA
ebprihar@wpi.edu

Neil Heffernan
ASSISTments.org
USA
neil@ASSISTments.org

Xintao Wu
University of Arkansas
USA
xintaowu@uark.edu

Ben Graff
Stride, Inc
USA
bgraff@k12.com

Dongwon Lee
Penn State University
USA
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 resource 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 discover that MathBERT pre-trained with 'mathVocab' outperforms MathBERT trained with the BASE BERT vocabulary (i.e., 'origVocab'). MathBERT is currently being adopted at the participated leaning platforms: Stride, Inc, a commercial educational resource provider, and ASSISTments.org, a free online educational platform. We release MathBERT for public usage at: https://github.com/tbs17/MathBERT.

CCS CONCEPTS

• **Applied computing** → **Education**; • **Computing methodologies** → **Natural language processing**.

KEYWORDS

BERT, Language Model, Mathematics Education, Text Classification

1 INTRODUCTION

The arrival of transformer-based language model, BERT [5], has revolutionized the NLP research and applications. One strength of BERT is its ability to adapt to new domain and/or task through pre-training by means of so-called "transfer learning." By taking an advantage of this benefit, therefore, researchers have adapted BERT into diverse domains (e.g., FinBERT [17], ClinicalBERT [11], BioBERT [13], SCIBERT [2], E-BERT [30], LiBERT [7]) and tasks (e.g., [27], [26], [3], [16], [8]) 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 improved context-aware understanding of language could help develop and improve solutions for challenging NLP tasks in mathematics.

In mathematics education, for instance, there are several general tasks that currently cause researchers/educators headaches: (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}). For instance, the struggle with T_{kc} (e.g., predicting the right mathematical skill for a given text description) is partly attributed to its tediousness and labor-intensive work for teachers/tutors to label all knowledge components in texts where they need to organize mathematical problems, or descriptions of instructional videos, etc. The traditional way to address this challenge of T_{kc} is to use machine learning to classify them via feature extraction [12, 19, 20], which has produced decent results.

However, open-ended essay or mathematical problem questions are becoming less popular in students' assignments due to the difficulty of developing universal automated support in assessing the response quality, causing educators to favor multiple choice questions when evaluating their students. According to Erikson et al. [6], from 2010 to 2020, less than 15% of the assigned open response

problems in ASSISTments [9] were ever graded by teachers. However, in general, open-ended questions are known to be able to provide critical evaluation in testing students' true critical thinking and understanding. Therefore, it is still important to develop an effective solution toward T_{kc}.

Similarly, *Knowledge Tracing*, a very important task in the education domain, is defined as the task of tracing students' knowledge state, which represents their mastery of educational content based on their past learning activities. Predicting students' next question correctness as a KT task is, for instance, well studied [4, 14, 15, 18, 28] but these solutions tend to rely on high-dimensional sequential data. The current solutions are still 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, therefore, in this work, we introduce our effort across two learning platforms (i.e., ASSISTments and K12.com) and three academic institutions (i.e., Penn State, WPI, and U. Arkansas) in the US: **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 light of the recent successes from transfer learning models such as ELMo [22], ULMFiT [10] and BERT [5], we propose to use a BERT-like model to improve the solutions of the aforementioned three tasks in one shot, as BERT has been proven to have outstanding performance in various NLP tasks.

However, directly applying BERT to mathematical tasks has limitations. First, the original BERT (i.e., BASE BERT) was trained mainly on general domain texts (e.g., general news articles and Wikipedia pages). As such, it is difficult to estimate the performance of a model trained on these texts on tasks using datasets that contain mathematical text. Second, the word distributions of general corpora is quite different from mathematical corpora (e.g., mathematical equations and symbols), which can often be a problem for mathematical task related models.

Therefore, we hypothesize that a special BERT model needs to be trained on mathematical domain corpora to be effective in mathematics-related tasks. 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. We publicly release MathBERT as a community resource at:
   - `https://github.com/tbs17/MathBERT` for codes on how to further-train and fine-tune, and
   - `https://huggingface.co/tbs17/MathBERT` for PyTorch version MathBERT and tokenizer.
   - AWS S3 URLs [1] for Tensorflow version MathBERT and tokenizer.

[1] http://tracy-nlp-models.s3.amazonaws.com/mathbert-basevocab-uncased/
http://tracy-nlp-models.s3.amazonaws.com/mathbert-mathvocab-uncased/

Table 1: Corpora Comparison for DAPT BERT Models

Domain	Name	# Tokens	Corpora
General NLP	Original BERT	3.3B	News article, Wikepedia
Bio Medicine	BioBERT	18B	PubMed, PMC articles
Clinical Medicine	ClinicalBERT	2M (notes)	Hospital Clinical Notes
Science	SciBERT	3.2B	Semantic Scholar Papers
Job	LiBERT	685M	LinkedIn search query profile, job posts
E-commerce	E-BERT	233M (reviews)	Amazon Dataset[2]
Finance	FinBERT	12.7B	Reuters News stories
Mathematics	**MathBERT** (This Work)	100M	Math curriculum and books, Math arXiv paper abstract

(2) We build and release a custom vocabulary `mathVocab` to reflect the different nature of mathematical corpora (e.g., mathematical equations and symbols). We compare the performance of MathBERT pre-trained with `mathVocab` to MathBERT pre-trained with the original BASE BERT vocabulary.
(3) We evaluate the performance of MathBERT for three general NLP tasks, T_{kc}, T_{ag} and T_{kt}, and compare its performance to five baseline models. Our experiments show that solutions of three tasks using MathBERT outperforms those using BASE BERT by 2-8%.
(4) We sketch the use cases of MathBERT currently being adopted at two major learning management systems: ASSISTments and K12.com by Stride.

2 RELATED WORK

The state-of-the-art language model BERT (Bidirectional Encoder Representations From Transformer) [5] is a pre-trained language representation model that was trained on 16 GB of unlabeled texts, including Books Corpus and Wikipedia, with a total of 3.3 billion words and a vocabulary size of 30,522. Its advantage over other pre-trained language models such as ELMo [22] and ULMFiT [10] is its bidirectional structure by using the *masked language model* (MLM) pre-training objective[5]. The MLM randomly masks 15% of the tokens from the input to predict the original vocabulary id of the masked word based on its context from both directions [5]. The pre-trained model can be used directly to fine-tune on new data for NLP understanding and inference tasks or further pre-trained to get a new set of weights for transfer learning.

The further pre-training process has become popular in the past two years as it is able to achieve better results than the fine-tuning only strategy. According to Gururangan et al. [8], there are two styles of further pre-training on the BASE BERT [5]: (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. [8] 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 al. [8] 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

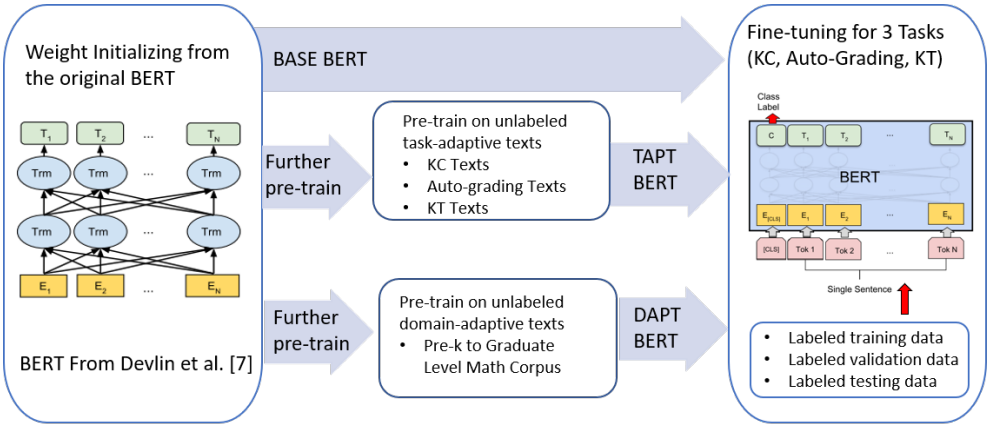

Figure 1: An illustration of training and fine-tuning process of BASE vs. TAPT vs. DAPT BERT models. The pre-training data are from this study. KC, Auto-grading, and KT Texts are task data for T_{kc}, T_{ag}, and T_{kc} respectively.

Table 2: Corpora Comparison for TAPT BERT Models. * indicates that the number is an estimation based on 150 tokens/sentence

Domain	Dataset	# Tokens	Task
BioMed	ChemProt [8]	1.5M*	relation classification
	RCT [8]	12M*	abstract sent. roles
Comp. Sci.	ACL-ARC [8]	291,150*	citation intent
	SCIERC [8]	697,200*	relation classification
News	HyperPartisan [8]	96,750*	partisanship
	AgNews [8, 27]	5.6M	topic
Reviews	Yelp [27]	25M	review sentiment
	IMDB [8, 27]	14.6M	review sentiment
Linguistics	VUA-20 [3]	205,425	metophor detection
	VUA-Verb [3]	5,873	metophor detection
Mathematics	KC [26]	589,549	skill code detection

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. 1).

The domain specific corpora that DAPT BERT models train on are usually huge (e.g. billions of news articles, clinical texts or PMC full-text and abstracts), which help DAPT BERT models achieve state-of-art (SOTA) performance in the corresponding domains. For example, FinBERT [17], ClinicalBERT [11], BioBERT [13], SCIBERT [2]. Other DAPT models such as E-BERT [30] and LiBERT [7] not only further pre-trained on the domain specific corpora but also modified the transformer architecture to achieve better performance for the domain related tasks. A comparison between different domain-specific BERT models' corpora is shown in Table 1. From the table, we can see that BioBERT was pre-trained on the largest set of tokens (18B) whereas our MathBERT is pre-trained on the smallest set of tokens (100M). Although the scale of training data is much smaller than the BASE BERT, MathBERT is still more effective in evaluating mathematics related tasks.

There are also a few works that focus on TAPT models. Sun et al. [27] proposed a detailed process on how to further pre-train a TAPT BERT model and fine-tune it for three types of classification tasks (i.e., sentiment, question, and topic), achieving a new record accuracy. Shen et al. [26] pre-trained a TAPT BERT model to predict knowledge components and surpassed the BASE BERT accuracy by about 2%. MelBERT [3] further pre-trained the RoBERTa-base BERT on well-known public English data sets (e.g.,VUA-20, VUA-Verb) that have been released in metaphor detection tasks and obtained [0.6%, 3%] out-performance over the RoBERTa-base [16]. Gururangan et al.[8] pre-trained RoBERTa-base [16] on famous task data sets (e.g., Chemprot, RCT, ACL-ARC, SCIERC, Hyperpartisan, AgNews, and IMDB tasks) and obtained [0.5%, 4%] better performance than RoBERTa-base. Table 2 presents the training data size for the aforementioned TAPT Models, showcasing that TAPT models have much smaller training data size than the DAPT BERT models. In general, DAPT models usually achieve better performance (1-8% higher) than TAPT models [8]. Although DAPT BERT models require more time and resource to train, they have wider applications than TAPT BERT models because they do not need to retrain in the case of different tasks, where TAPT BERT models tend to.

In light of the aforementioned success, we also build a DAPT model, MathBERT, that is further pre-trained from the BASE BERT model with a dedicated mathematical corpus. With the similar goal to our MathBERT, we note that the work by [21] was also independently announced about the same time (i.e., [21] was submitted to arXiv while our MathBERT was released to GitHub and Hugging Face, both in May 2021). [21] also built a pre-trained BERT from the mathematical formula data and applied it on three formula-related tasks (i.e., math info retrieval, formula topic classification, formula headline generation). However, as they claimed, their BERT is the first pre-trained model for mathematical formula understanding and was only trained on 8.7 million tokens of formula latex data with the 400 surrounding characters from arXiv papers (graduate-level). Our MathBERT is pre-trained on 100 million tokens of more general purpose mathematical corpora including curriculum, books, and arXiv paper abstracts, covering all the grade bands from pre-k to college graduate-level. Our training data not only include

Table 3: 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

formulas and their contexts but also more general mathematical instructional texts from books, curriculum, MOOC courses, etc. We consider our work has a potential to be widely used for "general" mathematics-related tasks. For instance, MathBERT in Hugging Face has been downloaded more than 150 times since May 2021. As [21] has not released their code and model artifacts, we could not compare our results directly to theirs. We welcome further comparison and analysis by releasing all our code and model artifacts at `https://github.com/tbs17/MathBERT`.

3 BUILDING MATHBERT

3.1 Math Corpora

MathBERT is pre-trained on mathematics related corpora that comprise 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 3). 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 all these data, the text book data is in PDF format and we hence converted them into text format using the Python package `pdfminer`[3], which preserves the mathematics symbols and equations (see sample text in Fig. 2).

3.2 Training Details and Outcome

To pre-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 [16]. In other words, one sequence of data is longer than the original single sentence from the mathematics corpora. Inspired by SciBERT [2], we create a custom mathematical vocabulary (`mathVocab`) using Hugging Face `BertWordPieceTokenizer`[4] with a size of 30,522 from the BASE BERT. 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 [5] (see in Table 4).

[3]https://pypi.org/project/pdfminer/
[4]https://huggingface.co/docs/tokenizers/python/latest/quicktour.html

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

We use 8-core TPU machine from Google Colab Pro to pre-train the BASE BERT on the mathematics corpora. The largest batch size (bs) we can fit into the TPU memory is 128 and the best training learning rate (lr) is $5e-5$ with maximum sequence length (max-seq) of 512 for both MathBERT with `origVocab` and `mathVocab`. We measure the effectiveness of training via Mask Language Modeling (MLM) accuracy (ACC), where the model predicts the vocabulary ID of the masked words in a sentence [5]. For training steps, we find both versions of MathBERT reach their best result at 600K with MLM accuracy of above 99.8% after a training time of 5 days each. We release MathBERT model artifacts trained with `origVocab` and `mathVocab` in both Tensorflow and Pytorch versions (see in `https://github.com/tbs17/MathBERT`). Specifically, one can use AWS S3 bucket URLs[5] to download the Tensorflow version of model artifact. The Pytorch version can be downloaded from the Hugging Face Repo[6] or directly installed within the Hugging Face's framework under the name space "`tbs17`" using the code below.

```
from transformers import AutoTokenizer
from transformers import AutoModelForMaskedLM
# Download the MathBERT-basevocab
tokenizer = AutoTokenizer.from_pretrained("tbs17/MathBERT")
model = AutoModelForMaskedLM.from_pretrained("tbs17/MathBERT")
# Download the MathBERT-mathvocab
tokenizer = AutoTokenizer.from_pretrained("tbs17/MathBERT-custom")
model = AutoModelForMaskedLM.from_pretrained("tbs17/MathBERT-custom")
```

[5]http://tracy-nlp-models.s3.amazonaws.com/mathbert-basevocab-uncased
http://tracy-nlp-models.s3.amazonaws.com/mathbert-mathvocab-uncased
[6]https://huggingface.co/tbs17/MathBERT

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

4 DOWNSTREAM MATH NLP TASKS

4.1 Three Tasks

We use three mathematical tasks mentioned in Section 1 to demonstrate the usefulness of MathBERT. They can be formulated as follows:

- 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).
- Auto-grading (T_{ag}): a two-sentence *multinominal classification* problem (5 labels) with $I \mapsto Question\&Answer$ pair and $O \mapsto Score$.
- KT Correctness (T_{kt}): a two-sentence *binary classification* problem with $I \mapsto Question\&Answer$ pair and $O \mapsto Correctness$.

4.2 Task Data

The three task data sets are noted as D_{kc} for T_{kc}, D_{ag} for T_{ag}, and D_{kt} for T_{kt}, respectively. They are used not only to fine-tune for task classification but also for pre-training TAPT BERT models,

Table 5: Task Data Details. KC: Knowledge Component, KT: Knowledge Tracing. All data from ASSISTments platform[9]

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 6: 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

which will serve as baseline models for MathBERT in Section 5. All of three data sets are provided from ASSISTments [9]. We use the same mathematical problem data set as in the best performing prior work [26] with 13,722 texts and 213 labels for KC prediction. The auto-grading task data is the same as in the best performing prior work [6] with 141,186 texts to predict scores 1 to 5. The KT data is the text version (269,230 texts and 2 labels) of the ASSISTments 2009 data[7], the numeric form of which was used by the best performing prior work [14].

Among the three data sets, D_{kc} has the smallest number of records (13,722 rows) but the most unique labels (213 labels), whereas D_{kt} has the largest number of records (269,230 rows) but the least unique labels (2 labels) (see in Table 5). These three data sets were chosen due to their accessibility and we don't expect our results would be significantly better or worse if we choose other data sets. When fine-tuning, both the labels and texts are used (see Column 2 and 3) with split ratio of 72% training, 8% validating, and 20% testing. When pre-training for TAPT BERT models, only the unlabeled texts are used for further pre-training without splitting (see Column 3).

Table 6 provides examples from the three task data sets. In D_{kc}, the label '8.EE.A.1' represents a knowledge component (KC) code where '8' means Grade 8, 'EE' is the skill name called 'Expression and Equation', and 'A.1' is the lesson code. There are total of 213 KC codes in D_{kc} with each represented by a specific knowledge component. In D_{ag}, the label '5' is the grading score '5' for the answer in the text. There are total of 5 labels in D_{ag} with '5' being the highest and '1' being the lowest. In D_{kt}, the label '1' means 'correct' for the answer in the text. There are total 2 labels in D_{kt} with another label '0' meaning 'incorrect' for student answers.

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

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	/
MathBERT+TAPT	T_{kc}	100K	100	99.99
	T_{ag}	100K	99.95	99.96
	T_{kt}	100K	99.67	99.68

4.3 Task Training and Fine-tuning

We pre-train BASE BERT on the unlabeled texts of D_{kc}, D_{ag}, D_{kt} to build TAPT BERT models and compare their performance to MathBERT. The difference between TAPT and DAPT BERT training is illustrated in Fig. 1 where the input corpora is different. DAPT BERT models have much larger corpora whereas TAPT BERT models are more specific to tasks. We pre-train three TAPT models with `origVocab` from the BASE BERT [5]. 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%. Each of the TAPT models takes approximately 1 day to train. In addition to creating TAPT models pre-trained from BASE BERT, we also pre-train TAPT models from the MathBERT weights, called MathBERT+TAPT. They reach the best results at steps of 100K for both `origVocab` and `mathVocab` with the MLM accuracy of above 99.6%. The MathBERT+TAPT models also take approximately 1 day each to pre-train. We try to keep the MLM accuracy of TAPT Models similar to MathBERT (see in Table 7).

For fine-tuning, we apply D_{kc}, D_{ag}, D_{kt} onto BASE BERT, TAPT BERT, MathBERT, and MathBERT+TAPT models separately. Below is an example code for fine-tuning on task data set with MathBERT weights and `origVocab`.

```
os.environ['TFHUB_CACHE_DIR'] = OUTPUT_DIR
python bert/run_classifier.py \
  --data_dir=$dataset \
  --bert_config_file=uncased_L-12_H-768_A-12_original/
     bert_config.json \
  --vocab_file=uncased_L-12_H-768_A-12_original/vocab.txt
      \
  --task_name=$TASK \
  --output_dir=$OUTPUT_DIR \
  --init_checkpoint=$MathBERT-orig_checkpoint \
  --do_lower_case=True \
  --do_train=True \
  --do_eval=True \
  --do_predict=True \
  --max_seq_length=512 \
  --warmup_step=200 \
  --learning_rate=5e-5 \
  --num_train_epochs=5 \
  --save_checkpoints_steps=5000 \
  --train_batch_size=64 \
  --eval_batch_size=32 \
  --predict_batch_size=16 \
  --tpu_name=$TPU_ADDRESS \
  --use_tpu=True
```

Table 8: 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

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, TAPT, and MathBERT+TAPT but vary from task to task. Table 8 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 $\in \{5, 10, 15, 25\}$.

5 EVALUATION OF MATHBERT

We denote MathBERT pre-trained with `origVocab` as MathBERT-orig and MathBERT pre-trained with `mathVocab` as MathBERT-custom. To evaluate their effectiveness across the tasks of T_{kc}, T_{ag} and T_{kt}, we fine-tune MathBERT on D_{kc}, D_{ag} and D_{kt} and compare the performance to the baseline models (see in Table 9). We group the baseline models into four categories: (1) Prior solutions with the best known performance, [6, 14, 26], (2) BASE BERT without any further pre-training, (3) TAPT BERT models pre-trained on the task specific texts from BASE BERT weights, and (4) MathBERT+TAPT models pre-trained on the task-specific texts from MathBERT weights in both `origVocab` and `mathVocab` versions.

We use both F1 and ACC (i.e., Accuracy) to measure T_{kc} prediction results because traditionally, KC problems have been evaluated using ACC [12, 19, 20, 25]. We provide the additional measure (F1) to account for the imbalance in the KC labels in D_{kc}. In addition, we use Area-Under-the-Curve (AUC) to measure T_{ag} because AUC is the typical measure used for the auto-grading problem. Finally, both AUC and ACC are used to measure T_{kt} because historically both metrics were used for evaluation [14, 18, 23, 31]. After obtaining the best hyper-parameter tuning for each task from Table 8, we run each model with five random seeds. We report the average value over five random seeds for each model and use t-tests to evaluate the significance of these results. A t-test is not applied to prior test results as we do not have the five random seeds results from the prior best method due to the lack of accessible codes.

In Table 8, we note that MathBERT-orig is about 1.38% to 22.01% better and MathBERT-custom is about 1.18% to 21.92% better than the best prior methods across all metrics and tasks. In addition, MathBERT-orig outperforms BASE BERT by about 2.14 % to 8.28%, all with statistical significance and MathBERT-custom outperforms it by about 1.98% to 8.21% across metrics and tasks, all with statistical significance. Both versions of MathBERT out-perform TAPT BERT models by [0.07%,0.98%] relatively with statistical significance for all tasks. We see both versions of MathBERT under-perform the MathBERT+TAPT models by 0.03 % to 1.77% across all the metrics except for F1 score on T_{kc} from MathBERT-orig. However, only the metrics for T_{kt} have obtained significance. This is expected as MathBERT+TAPT was further pre-trained by adapting it to the task-specific data on top of the MathBERT weights.

In addition, the best performance for each task is all from MathBERT related models. For example, for T_{kc}, the best F1 performance is from MathBERT-orig followed by the second best from MathBERT+TAPT-custom whereas the best and second-best ACC are from both of the MathBERT+TAPT versions (`origVocab`& `mathVocab`). For T_{ag}, we find the best AUC is from MathBERT+TAPT-orig followed by MathBERT-orig. For T_{kt}, the best and second best AUC and ACC are from both versions of MathBERT+TAPT with MathBERT+TAPT-custom having higher performance.

6 USE CASES

In this section, we describe the ongoing activities to incorporate MathBERT into two popular learning platforms.

6.1 ASSISTments

ASSISTments is an online learning platform that focuses on K-12 mathematics education. Within ASSISTments, teachers assign course work and view reports on their students. The reports show statistics on the class's performance and the responses of each student. Within the reports, 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.

These open ended responses provide the first opportunity to use MathBERT within ASSISTments. ASSISTments has recently begun using Sentence-BERT [24] to suggest grades to open response questions [1]. MathBERT provides a more domain-specific BERT model for this task with high AUC. The similar task in our experiment T_{ag} obtains 6.55% higher in AUC than the prior best work [6] which uses Sentence-BERT [1], and can replace the current Sentence-BERT implementation. MathBERT can not only provide teachers with suggested grades based on students' open ended responses, but also be used to suggest comments for teachers based on the content of the students' answers.

In addition to MathBERT's benefit to teachers using ASSISTments, MathBERT can also be used to enhance the student experience. As students complete problem sets in the ASSISTments Tutor, shown in Figure 4, they can be shown general educational material, such as YouTube videos, if they need additional guidance. MathBERT can be used to identify relevant content by predicting the skills required to solve the problem. As the fine-tuning results for T_{kc} using MathBERT-orig shows, the F1 score and ACC for the top 3 predictions are 92.67% and 93.79% respectively. Relevant supplemental education material can then be selected and shown to the student. Identifying the skills required to solve a problem will also integrate well with ASSISTments' Automated Reassessment and Relearning System (ARRS) [29]. This service automatically creates follow-up assignments for students when they fail to learn the material they were assigned. The purpose of the follow-up assignments is to test students' knowledge with problems similar to the ones the students previously got wrong. Although MathBERT was tested on text prediction tasks such as T_{kc}, T_{ag} and T_{kt}, it is not limited to only text prediction problems and can be applied to determine textual similarity, similar to the Semantic Textual Similarity Benchmark (STS-B) task from General Language Understand Evaluation

Table 9: Performance Comparison: MathBERT vs. Baseline Methods across Five Random Seeds. Bold font indicates best performance and underlined values are the second best. * indicates statistical significance. Δ shows relative improvement (%) of MathBERT over baselines.

Method	Vocab	T_{kc} (%)		T_{ag} (%)	T_{kt} (%)	
		F1	ACC	AUC	AUC	ACC
Prior Best (p)	/	88.69[26]	92.51[26]	85.00[6]	81.82[14]	77.11[14]
BASE-BERT (b)	orig	90.14	91.78	88.67	88.90	86.88
TAPT (t)	orig	91.77	92.96	90.34	95.88	93.49
MathBERT (m)	orig (o)	**92.67**	93.79	90.57	96.04	94.07
	math (c)	92.51	93.60	90.45	95.95	94.01
MathBERT+TAPT (mt)	orig (o)	92.54	93.82	**90.73**	97.25	95.52
	math (c)	92.65	**93.92**	90.46	**97.57**	**95.67**
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%***
Δ_{m-t}	orig	+0.98%***	+0.89%***	+0.25%**	+0.17%	+0.62%***
	math	+0.81%***	+0.69%***	+0.12%	+0.07%	+0.56%***
Δ_{m-mt}	orig	+0.14%	-0.03%	-0.18%	-1.26%***	-1.54%***
	math	-0.15%	-0.35%	-0.01%	-1.69%***	-1.77%***
$\Delta_{m^c-m^o}$	/	-0.17%	-0.20%	-0.13%	-0.09%	-0.06%
$\Delta_{mt^c-mt^o}$	/	+0.12%	+0.11%	-0.30%	+0.33%**	+0.16%

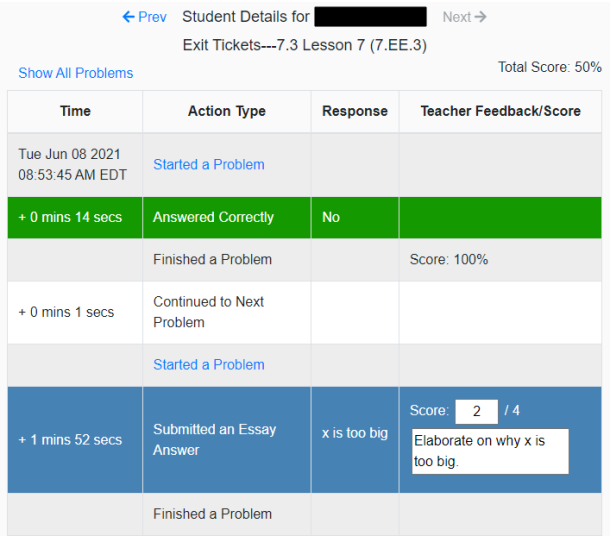

Figure 3: An open response in a student's report with the teacher's score and comment.

(GLUE)[8] which BASE BERT was evaluated on for its performance [5]. Therefore, we can use MathBERT to automatically evaluate problems for similarity, either by determining the skills required to solve the problems, or by directly comparing problem texts.

[8]https://gluebenchmark.com/

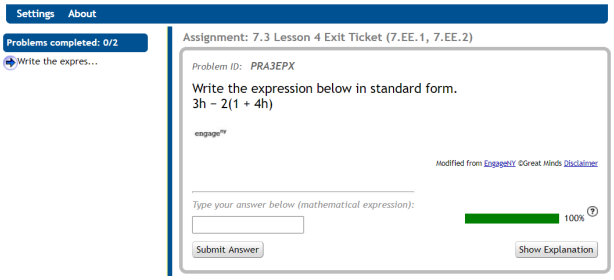

Figure 4: The ASSISTments Tutor, as seen by students when completing problem sets.

6.2 K12.com by Stride

Stride, Inc that manages the learning platform of K12.com, is a leading education management organization that provides online education to American students from kindergarten to Grade 12 as well as adults. K-G12 math teachers rely on the Stride system to give math lessons, assign practice, home work, or exams, and grade them to provide feedback to students. Teachers have long been challenged by the time and effort they spend to grade and give feedback on open-ended math questions where various answers could be right and it is difficult to scale feedback for immediacy and volume.

Therefore, Stride is considering an automatic scoring pipeline where they can train a model on their huge proprietary reservoir of open-ended responses and teacher feedback to automatically

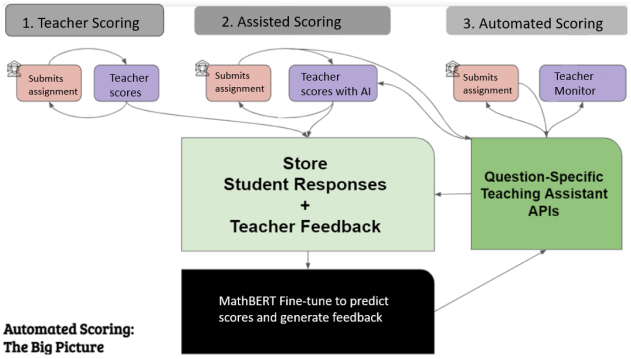

Figure 5: Stride auto-scoring pipeline using MathBERT

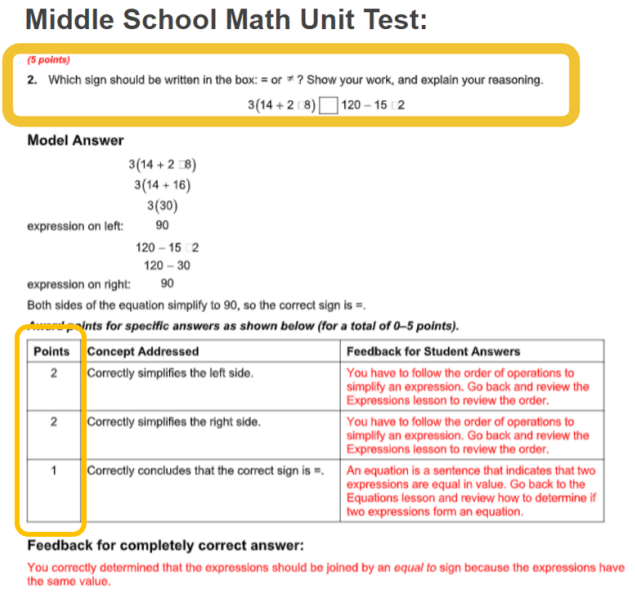

Figure 6: Stride auto-scoring model output in the unit test

suggest scores and generate constructive feedback/comments for teachers to use. MathBERT could be a nice fit for this model and play two roles: (i) MathBERT fine-tunes on students' responses (input) with ground truth teacher scoring (label) to predict scores with high accuracy (as suggested by T_{ag}), and (ii) MathBERT fine-tunes on the different scores (input) associated with teacher feedback (label) to predict/generate teacher feedback for a certain kind of score. For example, a student may only correctly answer part of the question and get a score of 3 out of 5, MathBERT can recommend a feedback such as 'You are very close! Can you tell us more?'. The prediction output from MathBERT can then be wrapped into a question-specific teaching assistant API that prompts in front of students to guide them to reach the full score and truly master the knowledge component (see the pipeline in Fig. 5).

The pipeline will be split into three phases: (i) collect data (i.e. responses, score, and feedback), (ii) use MathBERT to fine-tune on the training data and predict scores and feedback, suggested to teachers via API. Teachers semi-auto grade and give feedback using MathBERT suggested score and feedback. The final grade and feedback given to the students will then be sent back to the model to further fine-tine, and (iii) improve the accuracy of the question-specific teaching assistant API for fully automatic-scoring where teachers will only play a role in monitoring, reviewing the scores, and providing feedback.

As a proof of concept, Fig.6 illustrates what MathBERT will output after fine-tuning on the open-ended responses, scores, and feedback after phase 1. The red words are the feedback that the question-specific API will generate to guide students to achieve a full score. The points (in the yellow box) will be predicted by MathBERT and automatically suggested to teachers.

7 DISCUSSION AND LIMITATION

Although we have verified that MathBERT is more effective than the BASE BERT for mathematics related tasks with a proportional improvement of [1.98%, 8.28%] with statistical significance, the effect from an in-domain vocabulary (`mathVocab`) is not what we expect. As we see from Table 9, MathBERT-custom has underperformed MathBERT-orig when directly fine-tuned on, but outperformed MathBERT-orig when further pre-trained on task specific data. However, t-tests show MathBERT-orig is not significantly better than MathBERT-custom and MathBERT+TAPT-custom's out-performance over MathBERT+TAPT-orig is only statistically significant for T_{kc}.

As SciBERT [2] pointed out, the in-domain vocabulary is helpful but the out-performance over BASE BERT could be mainly from the domain corpus pre-training. Therefore, we argue that MathBERT trained with `mathVocab` sometimes can be more beneficial than MathBERT trained with `origVocab`. In addition, we note that MathBERT is not only applicable in text prediction tasks but also for other NLP understanding tasks such as paraphrasing, question and answering, and sentence entailment tasks. We evaluate MathBERT for T_{kc}, T_{ag}, and T_{kt} because three tasks have been heavily studied and their test data are available to us.

In future, we plan to pre-train another MathBERT on "informal" mathematics-related texts as opposed to the formal mathematical content (e.g. math curriculum, book and paper) that the current MathBERT is pre-trained on. We could potentially use such an informal MathBERT to generate answers/conversations for mathematics tutoring chat bots.

8 CONCLUSION

In this work, we built and introduced MathBERT-orig and MathBERT-custom to effectively fine-tune on three mathematics-related tasks. Users can use the code from github to access the model artifacts. We showed that MathBERT not only out-performed prior best methods by [1.18%, 22.01%], but also proportionally out-performed the BASE BERT by [1.98%, 8.28%] and TAPT BERT models by [0.25%, 0.98%] with statistical significance. MathBERT-custom was pre-trained with the mathematical vocabulary (`mathVocab`) to reflect the special nature of mathematical corpora and sometimes showed better performance than MathBERT-orig. MathBERT currently is

being adopted by two major learning management systems (i.e., ASSISTments and K12.com) to build automatic-scoring/commenting solutions to benefit teachers and students.

9 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] Sami Baral, Anthony F Botelho, John A Erickson, and Neil T Heffernan. 2021. Improving Automated Scoring of Student Open Responses in Mathematics. In *Educational Data Mining*.

[2] Iz Beltagy, Kyle Lo, and Arman Cohan. 2019. 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*. 3615–3620.

[3] Minjin Choi, Sunkyung Lee, Eunseong Choi, Heesoo Park, Junhyuk Lee, Dongwon Lee, and Jongwuk Lee. 2021. 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*.

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

[5] Jacob Devlin, Ming Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. 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*, Vol. 1. 4171–4186.

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

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

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

[9] Neil T. Heffernan and Cristina Lindquist Heffernan. 2014. 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 (2014), 470–497.

[10] Jeremy Howard and Sebastian Ruder. 2018. Universal Language Model Fine-tuning for Text Classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*. 328–339.

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

[12] Mario Karlovčec, Mariheida Córdova-Sánchez, and Zachary A. Pardos. 2012. 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 (2012), 195–200.

[13] Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. 2020. Data and text mining BioBERT: a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics* (2020), 1234–1240. https://doi.org/10.1093/bioinformatics/btz682

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

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

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

[17] Zhuang Liu, Degen Huang, Kaiyu Huang, Zhuang Li, and Jun Zhao. 2020. 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*.

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

[19] Zachary A Pardos. 2017. Imputing KCs with Representations of Problem Content and Context. In *Proceedings of the 25th Conference on User Modeling, Adaptation and Personalization*. 148–155. https://doi.org/10.1145/3079628.3079689

[20] Thanaporn Patikorn, David Deisadze, Leo Grande, Ziyang Yu, and Neil Heffernan. 2019. 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 (2019), 396–405.

[21] Shuai Peng, Ke Yuan, Liangcai Gao, and Zhi Tang. [n.d.]. MathBERT: A Pre-Trained Model for Mathematical Formula Understanding. In *arXiv preprint arXiv:2105.00377v1*.

[22] Matthew E Peters, Mark Neumann, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of NAACL-HLT*. 2227–2237.

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

[24] Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*. 3982–3992.

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

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

[27] Chi Sun, Xipeng Qiu, Yige Xu, and Xuanjing Huang. 2019. 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 (2019), 194–206.

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

[29] Yutao Wang and Neil T. Heffernan. 2014. The effect of automatic reassessment and relearning on assessing student long-term knowledge in mathematics. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 8474 LNCS (2014), 490–495.

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

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