A Compact Model for Mathematics Problem Representations Distilled from BERT

Hao Ming[1], Xinguo Yu[1,2*], Xiaotian Cheng[1,2], Zhenquan Shen[1], Xiaopan Lyu[1]

[1]Faculty of Artificial Intelligence in Education, Central China Normal University, Wuhan, China
[2]Central China Normal University Wollongong Joint Institute, Central China Normal University, Wuhan, China
{hming, xgyu}@ccnu.edu.cn, xiaotiancheng@mails.ccnu.edu.cn, {shenzhenquan, xiaopanlv}@ccnu.edu.cn

Abstract

Large language models (LLMs) have made significant advancements in math problem solving, but their large size and high latency render them impractical for real-world applications in intelligent mathematics solvers. Recently, task-agnostic compact models have been developed to replace LLMs in general natural language processing tasks. However, these models often struggle to acquire sufficient math-related knowledge from LLMs, leading to unsatisfactory performance in solving math word problems (MWPs). To develop a specialized compact model for representing MWPs, we develop the knowledge distillation (KD) technique to extract mathematical semantics knowledge from the large pre-trained model BERT. Effective knowledge types and distillation strategies are explored through extensive experiments. Our KD algorithm employs multi-knowledge distillation to extract fundamental knowledge from hidden states in the middle to lower layers, while also incorporating knowledge of mathematical relations and symbol constraints from higher-layer outputs and math decoder outputs, by leveraging bottleneck networks. Pre-training tasks on MWP datasets, such as masked language modeling and part-of-speech tagging, are also utilized to enhance the generalization of the compact model for MWP understanding. Additionally, a simple parameter mixing strategy is employed to prevent catastrophic forgetting of acquired knowledge. Our findings indicate that our approach can reduce the size of a BERT model by 10% while retaining approximately 95% of its performance on MWP datasets, outperforming the mainstream BERT-based task-agnostic compact models. The efficacy of each component has been validated through ablation studies.

Introduction

Solving Math Word Problems (MWPs) automatically has the potential to significantly advance both the development and application of intelligent math education. Large pre-trained language models (LLMs) (Peters et al. 2018; Devlin et al. 2019) have achieved remarkable success in many natural language processing tasks, benefiting from extensive parameters and training on large-scale corpora. Recently, LLMs have continuously improved their mathematical capabilities, and many of them have been successfully employed to develop high-quality MWP solvers (Qin et al. 2021), owing to their effective semantic representation and massive knowledge. However, due to the large parameter sizes and high latency, LLMs are often unsuitable for resource-constrained environments, such as the Internet of Things (IoTs) and small intelligent devices, which hinders their practical applications (Kok, Demirci, and Ozdemir 2024). Low-resource areas, especially in the education underdeveloped countries, have a greater need for lightweight models.

Knowledge distillation (Hinton, Vinyals, and Dean 2015) is an effective approach to model compression and has developed rapidly due to its efficacy in practical applications. Knowledge distillation involves transferring knowledge from a larger model (teacher) to a smaller one (student). Recently, task-agnostic knowledge distillation (Wang et al. 2020; Sun et al. 2020a) has been developed to construct compact models that can replace LLMs in general natural language processing tasks. These models are trained without task-specific data and do not require fine-tuning of the teacher model. However, student models often fail to learn sufficient knowledge from LLMs, and their performance on specific downstream tasks remains unsatisfactory even after fine-tuning. This issue may also arise in MWP solving, as the experimental results in this paper further confirm this inference. Successful attempts have been made to develop compact models in the fields of biomedicine and clinical applications (Rohanian et al. 2023). However, a compact model for MWP representations has not been reported yet, and no dedicated distillation strategy has been considered.

MWPs have unique symbols and linguistic representations of mathematics relationships, therefore, it is crucial to conduct task-specific distillation focused on mathematics representations and to design specialized knowledge types and distillation strategies. Within the studies of MWP solvers, it has been found that the part of speech (POS) and their combinations might encode underlying mathematical relations (Yu et al. 2023). Consequently, POS knowledge could be a fundamental basis for understanding MWPs. Inspired by this method, we propose a distillation approach that leverages POS pretraining and thoroughly explores the knowledge types and distillation strategies specifically for MWP representations. In this work, a compact model dedicated to MWP representation is proposed. The main contributions of this paper are summarized as follows:

*Corresponding authors.

- We are the first to focus specifically on creating a compact model for MWP representations through developing distillation technology.
- We investigate the essential knowledge types required for understanding MWPs and extract them from BERT by leveraging the bottleneck networks.
- An effective distillation strategy is designed to extract mathematics knowledge by integrating tailored pretraining and task-specific distillation techniques.
- On average, our compact model retains about 95% of BERT's performance on typical MWP datasets, while the parameters are only 10% of the original model.

Related Works

Math Word Problem Solving

Math word problem (MWP) solving involves deriving mathematics expressions and numerical solutions from a given problem text. Early research includes rule-based methods (Fletcher 1985; Bakman 2007), statistical machine learning methods (Kushman et al. 2014; Hosseini et al. 2014), and semantic parsing methods (Shi et al. 2015). Wang et al. (Wang, Liu, and Shi 2017) conducted a milestone study, introducing the Deep Neural Solver (DNS), the first neural solver for MWPs using a Seq2Seq structure. Following Wang's pioneering work DNS, deep learning-based methods have become mainstream in this research community due to their significant improvements in MWPs, including high accuracy and the elimination of handcrafted features. Deep learning-based MWP solvers typically utilize the Encoder-Decoder paradigm, where the encoder captures the semantics and relationships of the given problem text, and the decoder translates the encoder's outputs into mathematics equations with numbers and operators, ultimately deriving final answers.

Math expressions have natural hierarchical structures, so tree-structured solvers have often been designed to solve MWPs in recent studies. GTS (Xie and Sun 2019) is a goal-driven tree-structured neural model that generates an expression tree using a goal-driven mechanism; its decoder is widely used in MWP solvers. HMS (Lin et al. 2021) is a hierarchical math solver inspired by human reading habits. Xiong et al. (Xiong et al. 2022) proposed a variational information bottleneck to extract knowledge from the expression syntax tree. In these works, tree-structured decoders are employed to generate mathematics equations. The encoder's outputs directly influence the solver's performance, thus, LLM-based encoders can significantly improve the accuracy of MWP solving. For instance, MWP-BERT (Liang et al. 2022), DeductiveMWP (Jie, Li, and Lu 2022), and LogicSolver (Yang et al. 2022) use the BERT family (Devlin et al. 2019; Liu et al. 2019) as their encoder, while Gen&Rank (Shen et al. 2021) use BART. Additionally, MathBERT (Peng et al. 2021), and MWP-BERT, have been developed to enhance the capability of MWP solvers. However, these models overlook the high cost and latency in actual application environments, caused by the large size of LLMs. Math-solving engines are often deployed in lightweight and portable devices, where low resource usage and real-time response are essential in these applications.

Knowledge Distillation

Hinton first proposed the concept of knowledge distillation, which transfers knowledge from a large model (teacher) to a small model (student) using the loss of the soft target distributions. The goal of knowledge distillation is to minimize the differences between the teacher model and the student model:

$$\mathcal{L}_{KD} = \mathcal{L}(f^T(x), f^S(x)) \tag{1}$$

where $f^T(\cdot)$ and $f^S(\cdot)$ represent the features of teacher and student respectively, $\mathcal{L}$ is the loss function. Due to its great success in practical applications, knowledge distillation has become an effective approach for model compression. Task-specific distillation techniques typically fine-tune the teacher model on specific downstream tasks first and then have the student model emulate the teacher's behaviors, thereby achieving knowledge transfer from the teacher.

Knowledge types and distillation schemes are crucial to the effectiveness of distillation. Commonly used knowledge types can be divided into three categories (Gou et al. 2021). Response-based knowledge refers to the output response of the teacher's last layer, which is typically logits, also known as soft labels. It employs a method similar to label smoothing and has been widely used due to its simplicity and alignment with supervised learning. However, this knowledge type ignores the intermediate-level supervision provided by the teacher model. Feature-based knowledge primarily refers to the features of intermediate layers, namely feature maps. Feature maps contain more implicit knowledge and can help reduce the performance gap between the teacher and student. The selection of effective hint layers to guide the student model requires further exploration. Relation-based knowledge concerns the relationships between different layers or training data, such as the flow of solution process (FSP) matrix, multi-head graph (MHG), instance relations, and mutual information flow. Designing effective relation features also remains an open question. Distillation strategies (Park et al. 2024) include mutual distillation, adversarial distillation, muti-teacher distillation, data-free distillation, and self-distillation, etc. Various approaches have been proposed to transfer more effective knowledge and reduce the knowledge gap. For the distillation of LLMs, DistilBERT (Sanh et al. 2019), TinyBERT (Jiao et al. 2019), and MobileBERT (Sun et al. 2020b) are typical distilled BERT-base models and they are all evaluated on the General Language Understanding Evaluation (GLUE) benchmark. These approaches utilize a Transformer-like encoder to diminish the model gap with BERT, but differ in layer number and hidden size. Since distillation is conducted on general corpora, these models underperform on MWPs.

Dimension Reduction for MWP Representations

Reduced Vector Representation

As mentioned above, we know that BERT produces redundant vector representations when used for MWP solving, so we first need to determine how many dimensions of vector representation are sufficient for this task. Based on the solver

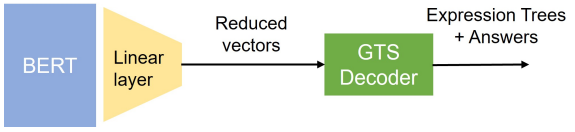

Figure 1: Vectors reduced from BERT using a linear layer are used for MWP solving.

Dimension	Acc. on Math23k	Acc. on Ape-clean
768	84.07%	84.33%
312	83.97%	84.12%
256	83.47%	83.62%
128	82.97%	80.23%

Table 1: The performance of reduced vector representation from BERT in solving MWPs.

architecture using BERT and GTS decoder, a linear network *Linear (m, n)* without activation is connected between them, as shown in Figure 1.

The hidden size of BERT is 768, so we set $m = 768$ and vary n from 768 to 128 to observe changes in solving accuracy. The results in Table 1 show that approximately 300 hidden states of BERT are sufficient for solving both Math23k (Wang, Liu, and Shi 2017) and Ape-clean datasets (Liang et al. 2022). This experiment indicates that the hidden states of BERT can be significantly compressed while retaining the ability of mathematics understanding, simply through a linear transform. Additionally, we infer that almost all reduced vector representations can retain task-specific capabilities by using the paradigm of linear layer and task-specific fine-tuning.

Bottleneck Network

The bottleneck network consists of two linear layers *Linear (m, n)* and *Linear (n, m)* stacked together. Inspired by the usage of linear layers for dimensionality reduction, we recognize that the bottleneck linear network is important for distilling a compact model from the teacher model when they have different hidden states. In this paper, we leverage this simple component to improve the distillation outcomes and efficiency. We add bottleneck linear networks to several designated intermediate layers of the teacher model when it is fine-tuned. This approach enables the effective transfer of the teacher's knowledge to the student model by reducing trainable parameters during the distillation stage.

Deep Distillation for Solving MWP

We focus on compressing the depth and width of the BERT model, which is more difficult than only compressing one of them.

Distillation Scheme Design

Model Architecture There are multiple distillation architectures, including multi-teacher distillation, distillation with a teacher assistant, mutual distillation, and self-distillation, etc. We focus on a distillation architecture that does not require excessive additional model components, distillation procedures, or datasets. Therefore, we choose to mine the knowledge from a single teacher to avoid the complexity of multi-teacher learning and utilize the teacher-student distillation paradigm. To eliminate the gap between the student model and teacher BERT, we employ the same transformer encoder architecture as the backbone of the student model, consisting of 3 encoder layers with a hidden size of 312. The feed-forward size is set to 1200, and the number of attention heads is set to 12. We use bottleneck linear networks to transfer the teacher's knowledge, the distillation mechanism is shown in Figure 2.

Fundamental Knowledge for MWP Understanding Various types of knowledge can be distilled from LLMs, however, there is no definitive guidance on which knowledge type is more effective (Hu et al. 2023) or how to integrate this knowledge for MWP solving. Part of speech (POS) is the fundamental knowledge for mathematics problem understanding, as some higher-level semantics are redundant for solving elementary math problems.

Considering a problem described as "*Q: Tom has four apples and two pears; how many fruits does he have in total?*" In this problem, the entities "apple" and "pear" could be replaced with other fruits without affecting the final solution. Therefore, the problem text representations generated by LLMs are redundant for MWPs, and constructing a lightweight encoder is a feasible alternative. As mentioned in the example above, we are not concerned with the specific names of some entities; we only need to ensure that they are all fruits and their mathematics relationships. Thus, POS represents an appropriate level of knowledge granularity that should be preserved in compact models. We also conducted experiments to verify this statement, the results are shown in Table 2, where the dataset used for this experiment is extracted from the problem text of Math23k and annotated using the LTP tool (Che et al. 2021), and $\text{BERT}_{\text{math23k}}$ represents the BERT full fine-tuned on Math23k. We know that the model will adjust its parameters when fine-tuned on the downstream tasks. It can be observed that BERT full fine-tuned on Math23k does not lose its ability to identify different POS types; in other words, solving MWPs requires retaining POS knowledge. We also conducted an experiment with the compact model to investigate the impact of POS knowledge. Table 3 demonstrates that the compact model exhibits improved performance on Math23k after pre-training with the POS tagging task. This further confirms that POS knowledge is fundamental to solving mathematics problems.

Keywords that represent mathematics relations are also crucial for MWP understanding. In the given question Q above, the keywords indicating addition operation are "*and*" and "*in total*." We can summarize these keywords to raise the attention of the model on them. In addition to POS and keywords, solving MWPs also demands an understanding of sentence structure and mathematics semantics. However, this knowledge is challenging to concretize, yet it can be extracted from specific layers of the teacher model through distillation techniques. Several studies indicate that the upper-

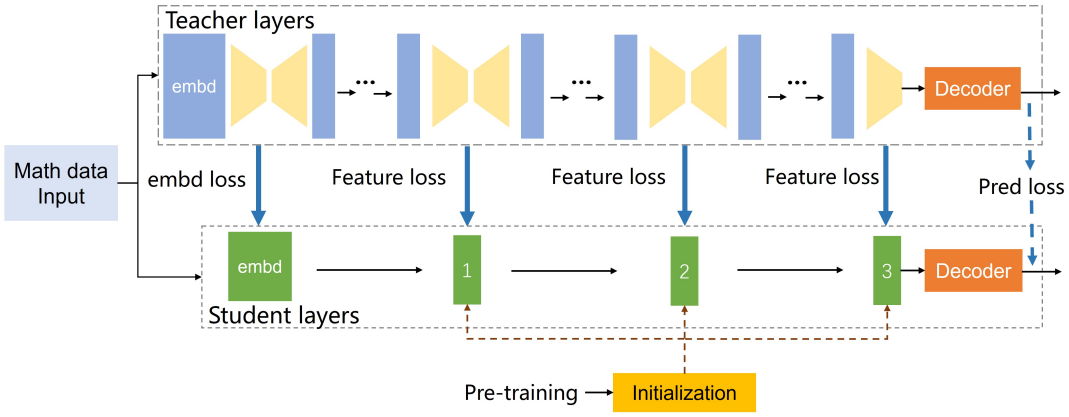

Figure 2: The proposed distillation mechanism for MWP representations operates on a 3-layer student network, including multiple knowledge extraction via the bottleneck layers.

Model	F1	Precision	Recall
$\text{BERT}_{\text{base}}$	79.51%	76.98%	82.22%
$\text{BERT}_{\text{math23k}}$	79.51%	76.98%	82.22%

Table 2: Performance of BERT on POS tagging with and without fine-tuning on math-solving tasks.

Model initialization	Equation Acc.	Answer Acc.
w/o POS tagging	45.59%	53.11%
w/ POS tagging	61.42%	71.15%

Table 3: The performance of the compact model on Math23k with and without POS tagging pretraining.

Layers	Equation Acc.	Answer Acc.	Perf. Prop. of BERT
1	51.00 %	60.52 %	72.08%
1-2	57.92%	67.84%	79.00%
1-3	59.21%	71.04%	80.79%
1-4	62.72%	73.55%	84.61%
1-6	66.83%	78.36%	87.59%
1-8	67.43%	79.46%	93.32%
1-10	69.64%	81.06%	94.63%
1-12	84.07%	84.33%	100%

Table 4: BERT with different layers is employed to evaluate mathematics problem-solving ability on Math23k.

middle layers of BERT are more effective than the top layer in guiding the student model. We aim to determine whether this phenomenon is observed in MWP tasks. We tested BERT with different layers to identify which layers are most effective, and the results are presented in Table 4.

It shows that the lower layers play important roles in MWP understanding. The first four layers account for 84.61% of the overall performance of the BERT model, with the first layer alone contributing 72.08%. These lower layers typically learn abstract knowledge such as morphology and syntax, underscoring the importance of fundamental knowledge for solving MWPs. The remaining eight layers account for the remaining 15.39% of performance. These layers, particularly the last few, usually learn task-specific knowledge, such as mathematics entity relationships in fine-tuned BERT. This phenomenon is consistent with the principle of diminishing marginal utility.

Distillation Procedure To achieve generalization and capture diverse knowledge, we follow the paradigm of combining pre-training with task-specific distillation. During the pre-training stage, the compact model learns elementary knowledge through Masked Language Modeling (MLM) and POS tagging tasks to enhance generalization. Subsequently, math task-specific distillation from the teacher

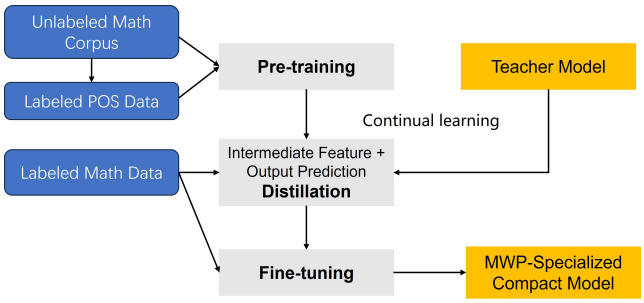

Figure 3: The proposed distillation scheme.

model equips the student model with MWP understanding capabilities. Figure 3 shows the proposed distillation procedure. To prevent catastrophic forgetting of the acquired knowledge, we employ a simple continual learning approach between pre-training and distillation. We adopt a straightforward method inspired by Wortsman (Wortsman et al. 2022), which involves assembling the weights of the pre-training and distillation models. We use this approach to yield two compact models at different stages, with the final model's weights derived by averaging the parameters of both. That is, $P_{final} = \theta P_{pretraing} + (1-\theta) P_{distillation}$.

Tom has four apples and two pears; how many fruits does he have in total?

Tom has [NUM] apples and [NUM] pears; how many fruits does he have in total?

Tom has Optional apples and Optional pears; Optional fruits does he have Optional?

Figure 4: Tailored MLM task for MWP understanding.

Pretraining Tasks for Basic Problem Understanding

The student model needs the right initialization for convergence and obtaining the elementary knowledge for solving MWPs. The traditional pretraining task is MLM, enabling the model to learn more semantic knowledge from large corpus. To understand MWPs more effectively, we pay more attention to the mask of mathematical keywords and numbers, these are the important components that distinguish MWPs from general problems. Figure 4 provides an example, where [NUM] is the number placeholder and underlined positions indicate that they can be substituted with masks.

We conduct MLM and POS tagging tasks on the large MWP dataset in the pretraining stage. The POS tagging datasets are obtained using the latest Chinese NLP tool called LTP. We preprocess the MWP datasets by extracting problem texts and performing POS tagging. In addition to Math23k and Ape210k, we also used HMWP (Qin et al. 2020) and CM17k (Qin et al. 2021) as the dataset during the pre-training phase. POS tagging requires word segmentation first, therefore we use the LTP tool to obtain the text after word segmentation and then use 'B-' to denote the beginning of words and 'I-' for words that continue. Next, we count the number of POS types in the datasets and add a classifier to the student model to predict these tags.

For MWPs, *nouns* (including temporal nouns), *quantity*, *verbs*, *punctuation*, and *pronouns* are important for understanding these problems. Among these, nouns might represent entities involved in mathematics operations, quantifiers might include unit conversions, and verbs could indicate mathematics operations. Focusing the student model on these POS types is fundamental to uncovering the key mathematics relations contained in the problem texts. The pre-training loss comprises the MLM task loss and the POS type prediction loss, with a 1:5 ratio between the two. We find that learning MLM and identifying POS from the MWP corpus can give the compact model the preliminary ability to solve MWPs.

Intuitively, some task-agnostic compact models have been pre-trained on large language corpus, hence we can use them to initialize our model. However, this paper focuses on discussing the effectiveness of our pre-training methods and does not utilize the knowledge of existing small models. Incorporating the knowledge from existing compact models can further enhance the capabilities of our model, a topic to be discussed in future work.

Task-specific Distillation

Intermediate Feature Distillation

Intermediate-layer knowledge plays an important role in distillation progress, which is crucial to eliminating the gap between student and teacher models. According to the literature (Liu et al. 2021), attention knowledge is not desirable information for distillation. Although the most attended token may contain important information, this may also hinder the student model from learning more crucial knowledge. For instance, [SEP] token may gain more attention, however, some trivial knowledge in its representation makes the attention distillation perform unsatisfactorily. Hidden states could contain rich semantic knowledge for better problem understanding, and implicit knowledge that might be used to solve the problems. We choose hidden states of intermediate layers for feature distillation. The loss function is:

$$\mathcal{L}_{Hidden} = \sum \alpha_n \mathcal{L}_H(H_l^S, H_{l'}^T W), \tag{2}$$

where H^S and H^T are the hidden states from the student and the teacher models, respectively. The symbols l and l' respectively denote the ith layer of the student model and the i'th layer of the teacher model. The loss function we use is mean squared error (MSE). α_n denotes the weight assigned to each loss. $W \in R^{d' \times d}$ is a linear matrix used to match the teacher's hidden size with that of the student. The layer mapping function is used to choose layers of the teacher model to match the student layers. We choose several layer mapping strategies, and their performance is summarized in Table 5 (S: student, T: teacher). The initial mapping is $S - T = \{(1, 4), (2, 8), (3, 12)\}$, where $S - T$ is a pair of numbered layers from student and teacher respectively. When a pair of layer mapping is changed, the other two pairs are retained.

Changed S-T layer mapping	Fixed S-T layer mapping	Equation accuracy	Answer accuracy
–	**1-4, 2-8, 3-12** (initialization)	**63.33 %**	**74.15 %**
1-1	2-8, 3-12	61.32 %	73.65 %
1-2	2-8, 3-12	61.52 %	73.95 %
1-3	2-8, 3-12	62.53 %	73.75 %
2-5	1-4, 3-12	60.82 %	71.74 %
2-6	1-4, 3-12	61.82 %	72.44 %
2-7	1-4, 3-12	61.92 %	74.05 %
3-9	1-4, 2-8	61.82 %	71.94 %
3-10	1-4, 2-8	61.12 %	72.34 %
3-11	1-4, 2-8	61.92 %	72.44 %

Table 5: Performance of the model with hidden-state distillation on Math23K, obtained under different layer mappings.

We observe that the student model performs better when its first layer learns from the initial four layers of the teacher model, and similarly, its performance improves when its final layer learns from the last layer of the teacher model. This aligns with our previous analysis regarding the distribution of knowledge types within the teacher model. Therefore, it is recommended that the student model primarily focuses on the information from the beginning and ending layers of the teacher model. Additionally, the middle layer of the student model should also learn from the middle layer

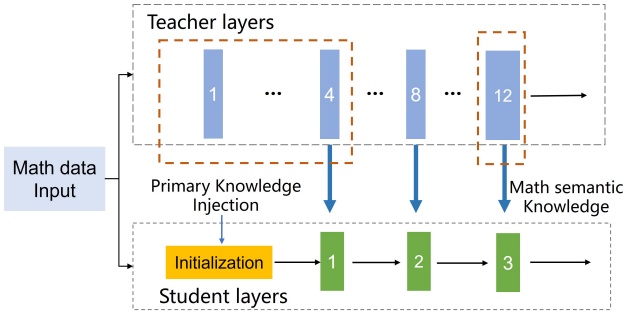

Figure 5: Extracting relevant knowledge for MWP solving from specific layers of the teacher model.

Knowledge type	Equation Acc.	Answer Acc.
attention matrix	58.72%	68.74%
FSP matrix	58.52%	68.44%
NST matrix	56.01%	66.43%
hidden states	63.33 %	74.15 %

Table 6: The impact of different knowledge transfers on distillation outcomes on Math23k dataset.

of the teacher model. We conjecture that this phenomenon is caused by the knowledge gap between different layers, that is, the student cannot learn the knowledge spanning too many teacher layers. Thus, we adopt a uniform mapping strategy for intermediate-layer distillation, that is, the mapping function is employed as:

$$l' = n \cdot (l_T / l_S) \tag{3}$$

where $n \in \{1, 2 \ldots, l_S\}$ and $\mathrm{mod}\,(l_T, l_S) = 0$, $l_S \leq l_T$. The distillation from intermediate layers utilizing a uniform mapping is illustrated in Figure 5. Since we pay more attention to the first and last layers, we assign weights of 1.0, 0.9, and 1.0 when calculating the hidden loss for the 3-layer compact model.

The result shows that the compact model can show good performance only by conducting intermediate-layer distillation from hidden states. There are other intermediate knowledge types, such as attention matrix, feature map, FSP matrix, etc. We also use these knowledge types for intermediate-layer distillation experiments. The result in Table 6 shows that relation-based knowledge is not suitable for solving MWP tasks.

We also distill the knowledge from the teacher's embedding layer and the loss function is:

$$\mathcal{L}_{Embd} = \mathcal{L}_H(E^S W_e, E^T), \tag{4}$$

where E^S and E^T are the embeddings of the student and the teacher models, respectively. We also take the MSE as the loss function, similar to hidden-layer distillation. Thus, the feature distillation is:

$$\mathcal{L}_{Fea} = \mathcal{L}_{Embd} + \mathcal{L}_{Hidden}. \tag{5}$$

Output Prediction Distillation

We also use the conventional distillation technology to distill output knowledge from the prediction layer of BERT. General distillation approaches distill logits (also called soft labels) of the last fully connected layer which usually is a classifier, and ground-truth labels (also called hard labels) are also used combined with soft labels. For MWP tasks, we employ the GTS decoder to generate output logits as the soft labels, often defined by a softmax function as:

$$p(z_i, T) = \frac{\exp(z_i/T)}{\sum_j \exp(z_j/T)}, \tag{6}$$

where z_i is the $i-th$ logit and T is the distillation temperature used to smooth the probabilities. The output of GTS can be simply regarded as the function of $w^T \tanh(\mathbf{W}[\mathbf{q}, \mathbf{c}, \mathbf{e}(y|P)])$, where $\mathbf{q}$ is a goal vector, $\mathbf{c}$ is a context vector containing the information of problem P, and $\mathbf{e}$ is the token embeddings of operators and quantities. GTS decoder can generate tree-structured outputs, thus it can be used to evaluate the gap between the generated expression tree and the equation annotated in datasets. The hard labels, namely the labeled solutions (expression trees) of MWP datasets are also used as the teacher knowledge. Thus, the prediction loss can be represented by:

$$\mathcal{L}_{Pred} = \lambda \mathcal{L}_{KD}\left(\frac{z_T}{T}, \frac{z_S}{T}\right) + (1-\lambda)\mathcal{L}_S(y, p(z_s)), \tag{7}$$

where z_T and z_S are the outputs from teacher and student models, respectively, λ is a hyperparameter. The second term $\mathcal{L}_S$ is often called the student loss, where y is the hard label. $\mathcal{L}_{KD}$ is the loss function, usually the cross-entropy loss. Thus, the total loss is the sum of the losses of each knowledge type:

$$\mathcal{L}_{Total} = \mathcal{L}_{Fea} + \mathcal{L}_{Pred}. \tag{8}$$

Comparison with Previous Work

Most existing works focus on task-agnostic compact models, aiming to reduce their size and replace general LLMs. These models have achieved good results in GLUE tasks. We compare our model with these compact models, and the result is presented in Table 7, where all models employ the GTS decoder to ensure fairness. L denotes the number of transformer layers and H represents the hidden size. TinyBERT applies Transformer distillation during both the pretraining and task-specific learning stages. TinyBERT$_4$ comprises 4 layers and achieves 96.8% of BERT's performance on the GLUE benchmark. DistillBERT is initialized with the teacher BERT's parameters and retains the same hidden size as the teacher. MiniLM distills only the last Transformer layer of the teacher model but comprises 6 layers with a hidden size of 384. Our model outperforms these mainstream compact models on math datasets while maintaining the smallest model size.

Experiments

Datasets

Our experiments mainly use four commonly used Chinese MWP datasets. Math23k is the most widely used dataset which contains 23162 math application problems with annotated equations and answers. Ape210k is an enormous

Models	Architecture	# Params	# FLOPs	Acc. on Math23k	Acc. on Ape-clean	Average
BERT (Teacher)	$L = 12, H = 768$	109M	22.5B	84.07%	84.33%	84.20%
DistilBERT	$L = 6, H = 768$	67M	11.3B	62.12%	75.71 %	68.92%
MiniLM	$L = 6, H = 384$	66M	-	58.67%	75.87%	67.27%
TinyBERT	$L = 4, H = 312$	14.5M	1.1B	55.41%	72.98%	64.12%
Our work	$L = 3, H = 312$	**10.3M**	**1.1B**	**80.23%**	**80.46%**	**80.35%**

Table 7: Comparison among the publicly released compact models distilled from BERT.

math dataset including 210488 MWPs. Since Ape210k has many noisy examples that miss annotations or cannot be solved, we use the re-organized datasets called Ape-clean (Liang et al. 2022) and full Ape210k can still be used for MLM pretraining. HMWP consists of 5470 MWPs including multi-unknown problems and non-linear problems, making problem solving more challenging. CM17K is another large-scale MWP dataset, which contains 6215 arithmetic problems, 5193 one-unknown linear problems, 3129 one-unknown nonlinear problems, and 2498 equation set problems. Because solving CM17K and HMWP needs another special decoder, we only use them in the pretraining stage. The solving performance of compact models is evaluated on Math23k and Ape-clean.

Implementation Details

We employ the fine-tuned version of Chinese pre-trained BERT with whole word masking (Cui et al. 2021) as the teacher model, which has a 12-layer Transformer encoder with 768 hidden states and 12 attention heads. Our model is implemented by PyTorch on an NVIDIA A800 100 GB GPU. At the pretraining stage, 150 epochs are trained using the Adam optimizer with the initial learning rate of 1e-5 and weight decay of 1e-5, the mini-batch size is set to be 128. At the task-specific distillation stage, we also use the Adam optimizer, and the initial learning rate is set as 3e-5, and we pre-train them 120 epochs. The loss weights α_1, α_2, and α_3 of our distillation loss obtained by grid search are set as 1.0, 0.9, and 1.0. According to our extensive experiments, the hyperparameter θ and λ are set as 0.2 and 0.5, respectively. The temperature factor for soft labels is set to 4. We fine-tune the student model using a batch size of 32 for 100 epochs, and the dropout rate is 0.1. The initial fine-tuning learning rate is set as 1e-5 and 1e-4 for the student model and GTS, respectively. Other compact models are implemented according to their specific settings in previous literature.

Ablation Studies

We conduct ablation studies to analyze the contributions of main components in the distillation scheme. We mainly focus on different procedures of the proposed distillation and different distillation knowledge. The Results are presented in Table 8. It indicates that pretraining tasks are crucial to enhancing the efficacy of the proposed method, and the POS tagging task positively contributes to the pretraining procedure. In terms of the proposed distillation objectives, all of them contribute to model performance, with intermediate-layer distillation proving more beneficial in task-specific distillation. We also investigate the contributions of soft and hard labels and find that hard labels contribute almost as much as soft labels. These two types of knowledge from the prediction layer are complementary to each other and are both important for the final distillation results.

Model	Math23k	Ape-clean
w/o Pretraining	77.5%	78.4%
w/o POS tagging	78.1%	79.0%
w/o MLM	78.9%	79.6%
w/o Hidden Distillation	76%	77.4%
w/o Prediction Distillation	78.6%	79.1%
w/o soft label	79.2%	79.4%
w/o hard label	79.3%	79.8%

Table 8: Ablation studies of different components.

Conclusion and Future Work

In this paper, we find that the representations of task-agnostic compact models are inadequate for solving MWPs. Thus, we propose a new compact model to facilitate the practical application of intelligent mathematics solvers. Pretraining and multi-knowledge distillation are utilized for math-related knowledge extraction and progressive transfer. Empirical results on commonly used MWP datasets demonstrate that our model achieves performance comparable to BERT, while the size of the model layer and hidden states can be much smaller. Extensive experiments reveal that 1) POS knowledge and hidden states are important for solving MWPs, 2) the uniform mapping principle is still effective for layer-level knowledge transfer, and 3) the compact model can benefit from both our tailored pretraining and distillation. This is the first work to develop a math-domain compact model, and we believe our model can facilitate both research and practical applications in MWP solving.

In future work, we will further refine the knowledge of mathematical relations within problems, utilizing a larger math-related corpus and a more robust teacher model for guidance. Concurrently, we will optimize the knowledge distillation techniques, including reducing the steps involved in distillation and employing smaller-scale models. We can use existing task-agnostic compact models as the initialization for our model to enhance generalization performance while reducing training costs. Additionally, we will explore the potential of other types of LLMs, such as GPT, in constructing compact models for solving math problems.

Acknowledgments

This work is partially supported by the General Program of the National Natural Science Foundation of China (Grant No: 62277022) and the China Postdoctoral Science Foundation (Grant No: 2023M731245).

References

Bakman, Y. 2007. Robust understanding of word problems with extraneous information. arXiv:0701393.

Che, W.; Feng, Y.; Qin, L.; and Liu, T. 2021. N-LTP: An Open-source Neural Language Technology Platform for Chinese. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, 42–49.

Cui, Y.; Che, W.; Liu, T.; Qin, B.; and Yang, Z. 2021. Pre-Training With Whole Word Masking for Chinese BERT. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 29: 3504–3514.

Devlin, J.; Chang, M.; Lee, K.; and Toutanova, K. 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*, 4171–4186.

Fletcher, C. R. 1985. Understanding and solving arithmetic word problems: A computer simulation. *Behavior Research Methods, Instruments, & Computers*, 17(5): 565–571.

Gou, J.; Yu, B.; Maybank, S. J.; and Tao, D. 2021. Knowledge distillation: A survey. *International Journal of Computer Vision*, 129(6): 1789–1819.

Hinton, G.; Vinyals, O.; and Dean, J. 2015. Distilling the knowledge in a neural network. arXiv:1503.02531.

Hosseini, M. J.; Hajishirzi, H.; Etzioni, O.; and Kushman, N. 2014. Learning to Solve Arithmetic Word Problems with Verb Categorization. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP*, 523–533.

Hu, C.; Li, X.; Liu, D.; Wu, H.; Chen, X.; Wang, J.; and Liu, X. 2023. Teacher-Student Architecture for Knowledge Distillation: A Survey. arXiv:2308.04268.

Jiao, X.; Yin, Y.; Shang, L.; Jiang, X.; Chen, X.; Li, L.; Wang, F.; and Liu, Q. 2019. Tinybert: Distilling bert for natural language understanding. arXiv:1909.10351.

Jie, Z.; Li, J.; and Lu, W. 2022. Learning to Reason Deductively: Math Word Problem Solving as Complex Relation Extraction. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics , ACL*, 5944–5955.

Kok, I.; Demirci, O.; and Ozdemir, S. 2024. When IoT Meet LLMs: Applications and Challenges. arXiv:2411.17722.

Kushman, N.; Zettlemoyer, L.; Barzilay, R.; and Artzi, Y. 2014. Learning to Automatically Solve Algebra Word Problems. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, ACL*, 271–281.

Liang, Z.; Zhang, J.; Wang, L.; Qin, W.; Lan, Y.; Shao, J.; and Zhang, X. 2022. MWP-BERT: Numeracy-Augmented Pre-training for Math Word Problem Solving. In *Findings of the Association for Computational Linguistics: NAACL*, 997–1009.

Lin, X.; Huang, Z.; Zhao, H.; Chen, E.; Liu, Q.; Wang, H.; and Wang, S. 2021. HMS: A Hierarchical Solver with Dependency-Enhanced Understanding for Math Word Problem. In *Thirty-Fifth AAAI Conference on Artificial Intelligence, EAAI*, 4232–4240.

Liu, Y.; Meng, F.; Lin, Z.; Wang, W.; and Zhou, J. 2021. Marginal Utility Diminishes: Exploring the Minimum Knowledge for BERT Knowledge Distillation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP*, 2928–2941. Association for Computational Linguistics.

Liu, Y.; Ott, M.; Goyal, N.; Du, J.; Joshi, M.; Chen, D.; Levy, O.; Lewis, M.; Zettlemoyer, L.; and Stoyanov, V. 2019. RoBERTa: A Robustly Optimized BERT Pretraining Approach. arXiv:1907.11692.

Park, S.; Choi, J.; Lee, S.; and Kang, U. 2024. A comprehensive survey of compression algorithms for language models. arXiv:2401.15347.

Peng, S.; Yuan, K.; Gao, L.; and Tang, Z. 2021. MathBERT: A Pre-Trained Model for Mathematical Formula Understanding. arXiv:2105.00377.

Peters, M. E.; Neumann, M.; Iyyer, M.; Gardner, M.; Clark, C.; Lee, K.; and Zettlemoyer, L. 2018. Deep Contextualized Word Representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT*, 2227–2237.

Qin, J.; Liang, X.; Hong, Y.; Tang, J.; and Lin, L. 2021. Neural-Symbolic Solver for Math Word Problems with Auxiliary Tasks. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP*, 5870–5881.

Qin, J.; Lin, L.; Liang, X.; Zhang, R.; and Lin, L. 2020. Semantically-Aligned Universal Tree-Structured Solver for Math Word Problems. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP*, 3780–3789.

Rohanian, O.; Nouriborji, M.; Kouchaki, S.; and Clifton, D. A. 2023. On the effectiveness of compact biomedical transformers. *Bioinformatics*, 39(3).

Sanh, V.; Debut, L.; Chaumond, J.; and Wolf, T. 2019. DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. arXiv:1910.01108.

Shen, J.; Yin, Y.; Li, L.; Shang, L.; Jiang, X.; Zhang, M.; and Liu, Q. 2021. Generate & Rank: A Multi-task Framework for Math Word Problems. In *Findings of the Association for Computational Linguistics: EMNLP*, 2269–2279.

Shi, S.; Wang, Y.; Lin, C.; Liu, X.; and Rui, Y. 2015. Automatically Solving Number Word Problems by Semantic

Parsing and Reasoning. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, EMNLP*, 1132–1142.

Sun, Z.; Yu, H.; Song, X.; Liu, R.; Yang, Y.; and Zhou, D. 2020a. MobileBERT: a Compact Task-Agnostic BERT for Resource-Limited Devices. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL*, 2158–2170. Association for Computational Linguistics.

Sun, Z.; Yu, H.; Song, X.; Liu, R.; Yang, Y.; and Zhou, D. 2020b. Mobilebert: a compact task-agnostic bert for resource-limited devices. arXiv:2004.02984.

Wang, W.; Wei, F.; Dong, L.; Bao, H.; Yang, N.; and Zhou, M. 2020. MiniLM: Deep Self-Attention Distillation for Task-Agnostic Compression of Pre-Trained Transformers. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems, NeurIPS*.

Wang, Y.; Liu, X.; and Shi, S. 2017. Deep Neural Solver for Math Word Problems. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, EMNLP*, 845–854.

Wortsman, M.; Ilharco, G.; Kim, J. W.; Li, M.; Kornblith, S.; Roelofs, R.; Lopes, R. G.; Hajishirzi, H.; Farhadi, A.; Namkoong, H.; et al. 2022. Robust fine-tuning of zero-shot models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 7959–7971.

Xie, Z.; and Sun, S. 2019. A Goal-Driven Tree-Structured Neural Model for Math Word Problems. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI*, 5299–5305.

Xiong, J.; Li, C.; Yang, M.; Hu, X.; and Hu, B. 2022. Expression Syntax Information Bottleneck for Math Word Problems. In *SIGIR '22: The 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2166–2171.

Yang, Z.; Qin, J.; Chen, J.; Lin, L.; and Liang, X. 2022. LogicSolver: Towards Interpretable Math Word Problem Solving with Logical Prompt-enhanced Learning. In *Findings of the Association for Computational Linguistics: EMNLP*, 1–13.

Yu, X.; Lyu, X.; Peng, R.; and Shen, J. 2023. Solving arithmetic word problems by synergizing syntax-semantics extractor for explicit relations and neural network miner for implicit relations. *Complex & Intelligent Systems*, 9(1): 697–717.