MWP-BERT: A Numeracy-augmented Pre-trained Encoder for Math Word Problems

Zhenwen Liang[1], **Jipeng Zhang**[2], **Lei Wang**[3],
Wei Qin[4], **Jie Shao**[5], and **Xiangliang Zhang**[✉1]

[1]University of Notre Dame, {zliang6, xzhang33}@nd.edu
[2]Hong Kong University of Science and Technology, jzhanggr@conect.ust.hk
[3]Singapore Management University, lei.wang.2019@phdcs.smu.edu.sg
[4]Hefei University of Technology, qinwei.hfut@gmail.com
[5]University of Electronic Science and Technology of China, shaojie@uestc.edu.cn

Abstract

Math word problem (MWP) solving faces a dilemma in number representation learning. In order to avoid the number representation issue and reduce the search space of feasible solutions, existing works striving for MWP solving usually replace real numbers with symbolic placeholders to focus on logic reasoning. However, instead of the number value itself, it is the reusable numerical property that matters more in numerical reasoning. Therefore, we argue that injecting numerical properties into symbolic placeholders with contextualized representation learning schema can provide a way out of the dilemma in the number representation issue here. In this work, we introduce this idea to the popular pre-training language model (PLM) techniques and build MWP-BERT, an effective contextual number representation PLM. We demonstrate the effectiveness of our MWP-BERT on MWP solving and several MWP-specific understanding tasks on both English and Chinese benchmarks.

1 Introduction

MWP solving system aims to perform symbolic reasoning by searching through a combinatorial solution space given the text description evidence. Despite the great performance achieved by the previous methods, there still exists fundamental challenges in number representation for MWP solving. More exactly, number values are required to be considered as vital evidence in solution exploration but existing works are known to be inefficient in capturing numeracy information Wallace et al. [2019]. Intuitively, we could simply treat explicit numbers in the same way with words, i.e., assign position for all numbers in the vocabulary. However, there would be an infinite number of candidates during prediction and it would be impossible to learn their deep representations. In other words, the solution space will be extremely large and the complexity is unacceptable. Therefore, almost all existing works follow the number mapping technique Wang et al. [2017] to replace all numbers with symbolic placeholders (e.g., "x1", "x2"). The core idea here is to get a reasonable solution space by restricting neural networks to leave out numerical characteristics and focus on logic reasoning. However, most of the current MWP solvers do not consider the background knowledge in the context and are usually inefficient in capturing numeracy properties. An example is shown in Fig. 1. Small perturbations in the problem description actually bring large variations in reasoning logic and equation. If the model simply regards "75" and "10%" as the same placeholder "x3", and does not notice the small variation in the context, a wrong solution will be generated.

To this end, a group of numeracy grounded pre-training objectives is designed to leverage the corpus of MWP and encourage the contextual representation to capture numerical information.

36th Conference on Neural Information Processing Systems (NeurIPS 2022) Workshop on Math-AI.

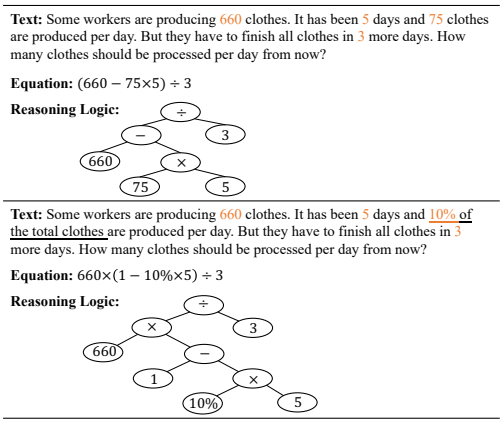

Figure 1: The second question is obtained from the first one by minor modifications. However, their solution equation and corresponding equation tree structure are different from each other.

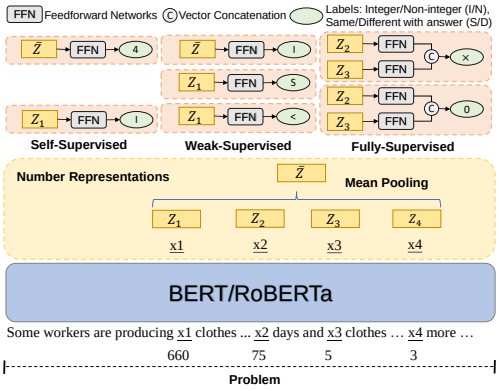

Figure 2: The overall architecture of our BERT-based MWP solver. Our method enables the solver to learn from unlabeled, incompletely labeled and fully labeled MWPs by different pre-training tasks.

Experiments conducted on both Chinese and English benchmarks show the significant improvement of our proposed approach over all competitors.

2 Method

An overview of pre-training objectives and our model architecture is shown in Figure 1. In general, pre-training objectives are designed to inject contextual priori and numerical properties as soft constraints for representation learning. They are categorized into three types given provided training signals, i.e., self-supervised, weakly-supervised, and fully-supervised.

2.1 Self-supervised Objectives

In this part, we only consider input text descriptions for each example. Also, these objectives can alleviate the costs of collecting MWP corpus by constructing supervision signals without solution answers and equations.

Masked Language Modeling. We follow Devlin et al. [2019] and introduce masked language modeling (MLM) for basic contextual representation modeling. Specially, we apply masks on 10% of tokens, randomly replace 10% of tokens with other tokens and keep 80% of tokens unchanged. Later, the manipulated sentence is utilized to reconstruct the original sentence.

Number Counting. Another pre-training objective is to predict the number of numbers that appeared in MWP description. The amount of a number corresponds with the cardinality of variable sets. This also reflects the basic understanding of the difficulty of an MWP and can act as a key contextual MWP number understanding feature.

Number Type Grounding. This objective aims at linking contextual number representations with corresponding number types to tell the difference between discrete and continuous concepts/entities. For numerical reasoning in MWP solving, we only need to handle whole numbers as well as non-integer numbers (decimal, fraction and percentage). Ideas here are that whole numbers usually associate with discrete entities (for example, desks, chairs and seats) while non-integer numbers often connect with continuous concepts (for example, proportions, rate, velocity). Besides, comparisons among whole numbers got different issues compared with rational numbers. Therefore, we propose a classification objective to predict if a number is a whole number or a non-integer number.

2.2 Weakly-supervised Objectives

Given both text descriptions of MWPs and corresponding value answers, we can model dependencies among answer number and numbers in text descriptions so that contextual representation perceive the existence of the target variable number that does not appear in the text descriptions. In detail, we design 3 novel pre-training objectives specializing in value-annotated MWPs to improve number representation in our MWP-BERT.

Answer Type Prediction. Determining the type of answer number can provide us discrete/continuous nature of the target entity/concept. Thus, we want to predict the type (whole/non-integer) of the answer value given global representations of an MWP.

Context-Answer Type Comparison. Besides the global context feature, an MWP-BERT also needs to associate context numbers and answer numbers (the target number does not explicitly appear in the text). Thus, another objective is proposed to predict if the quantities appeared in the MWP text fall into the same category as the answer (i.e. they are all whole or non-integer).

Number Magnitude Comparison. Beyond type, the magnitude of a number serves as the foundation of numerical reasoning. By associating magnitudes evaluation with contextual representation, the model can get a better perception of variance over key reasoning cues like time, size and intensity.

2.3 Fully-supervised Objectives

Given both equations and answers for MWPs, we can design fully-supervised training tasks to associate number representation with reasoning flows (solution equation). Mathematical equations are known to be binary tree structures with operators on root nodes and numbers on leaf nodes. The motivation is to encourage models to learn structure-aware number representations that encode the information on how to make combinations over atomic operators and numbers. We incorporate two pre-training objectives based on the solution equation tree.

Operation Prediction. The first one is a quantity-pair relation prediction task that focuses on the local feature of the equation tree. The goal is to predict the operator between two quantity nodes in the solution tree. This is in fact a classification task with 5 potential targets, i.e., $+, -, \times, \div$ and $\wedge$.

Tree Distance Prediction. Another pre-training objective is to incorporate the global structure of the equation tree in a quantitative way. Inspired by Hewitt and Manning [2019], we consider the depth of each number and operator on the corresponding binary equation tree to be the key structure priori. Thus, we design another fully-supervised objective to utilize this information. More exactly, given the representation of two number nodes in an equation tree, this is a regression problem that predicts the distance (difference of their depth) between them.

3 Experiment

For the Chinese initial model, we use an upgraded patch of Chinese BERT which is pre-trained with the whole word masking (WWM)[1] Cui et al. [2020]. For the English pre-training models, we use the official source on this website[2].

3.1 Dataset

We conduct experiment based on Math23k Wang et al. [2017], MathQA Amini et al. [2019] and Ape-210k Zhao et al. [2020]. Since many noisy examples exist in Ape-210k, e.g., examples without equation annotations or answer values, we re-organize Ape-210k to Ape-clean and Ape-unsolvable, where the training set of Ape-clean and the whole Ape-unsolvable are used for pre-training. For the English MWP, we use the training set of MathQA Amini et al. [2019] to perform pre-training.

[1] `https://github.com/ymcui/Chinese-BERT-wwm`

[2] `https://huggingface.co/bert-base-uncased` and `https://huggingface.co/roberta-base`

3.2 Probing Evaluation

We re-run all the pre-training tasks as probing tasks to evaluate our modeling's understanding ability and test MWP-BERT in a zero-shot scenario, i.e. without fine-tuning the parameters of MWP-BERT and MWP-RoBERTa for the sake of fair comparison. Besides, we borrow an MWP-specific sequence labeling task, quantity tagging Zou and Lu [2019] ("QT"), to further compose MWP understanding evaluation settings. Briefly speaking, this task requires the model to assign "+", "-" or "None" for every quantity in the problem description and can serve as an MWP understanding evaluation tool to examine the model's understanding of each variable's logic role in the reasoning flow. We extract the corresponding vectors of all quantities according to their positions in the encoded problem. Next, a 2-layer feed-forward block is connected to output the final prediction. Significant improvements can be observed in Table 1, and demonstrate the effectiveness of our proposed pre-training techniques in improving the number representation of PLMs.

	NumCount	NTGround	ATPred	CATComp	NumMComp	OPred	TPred	QT
Metric	MSE ↓	Acc ↑	Acc ↑	Acc ↑	Acc ↑	Acc ↑	MSE ↓	Acc ↑
BERT	3.08	0.87	0.75	0.77	0.77	0.50	0.97	84.5
RoBERTa	3.20	0.86	0.76	0.78	0.77	0.51	0.99	84.6
MWP-RoBERTa	0.69	0.92	0.86	0.87	0.86	0.86	0.44	91.0
MWP-BERT	0.67	0.92	0.85	0.87	0.86	0.87	0.45	91.5

Table 1: The evaluation results on MWP-specific understanding tasks. All tasks correspond to the tasks mentioned in section 1. Note that the metric for 2 tasks is mean-squared-error, while others use classification accuracy. "QT" stands for quantity tagging.

3.3 MWP Solving

Given a textual description of a mathematical problem, which contains several known variables, MWP solving targets getting the correct answer for the corresponding question. A solver is expected to be able to predict an equation that can exactly reach the answer value. We adapt our proposed encoder with multiple different traditional solvers by replacing their RNN encoder with MWP-BERT to show its generalization ability across various solvers. The results show that our MWP-BERT outperforms vanilla BERT Devlin et al. [2019] and has great adaptivity on different solvers and we successfully achieve state-of-the-art accuracy.

	Math23k	Math23k*	MathQA
State-of-the-art Baselines			
REAL Huang et al. [2021]	82.3	80.0	–
BERT-CL Li et al. [2021]	83.2	–	76.3
Gen&Rank Shen et al. [2021]	85.4	**84.3**	–
DeductiveReasoner Jie et al. [2022]	85.1	83.0	78.6
Adapting MWP-BERT			
BERT Devlin et al. [2019] + GTS Xie and Sun [2019]	83.8	82.0	75.1
MWP-BERT + GTS Xie and Sun [2019]	84.7	82.4	76.2
MWP-BERT + Teacher Liang and Zhang [2021]	85.1	82.8	77.3
MWP-BERT + Graph2Tree Zhang et al. [2020b]	**85.6**	83.8	**78.9**

Table 2: Comparison of answer accuracy (%) among our proposed models and different baselines. Math23k column shows the results on the public test set and Math23k* is 5-fold cross validation on Math23k dataset. MathQA is adapated from Li et al. [2021], Tan et al. [2021]. "BERT" represent results without our pre-training.

4 Conclusion

We propose MWP-BERT, an MWP-specific PLM model with 8 pre-training objectives to solve the number representation issue in MWP. Experimental results show the superiority of our proposed MWP-BERT across various downstream tasks on generation and understanding. In terms of the most representative task MWP solving, our approach achieves state-of-the-art. Better numerical understanding ability is also demonstrated in the probing evaluation. We believe that our study can serve as a useful pre-trained pipeline and a strong encoder in the MWP community.

References

Aida Amini, Saadia Gabriel, Shanchuan Lin, Rik Koncel-Kedziorski, Yejin Choi, and Hannaneh Hajishirzi. Mathqa: Towards interpretable math word problem solving with operation-based formalisms. In *NAACL*, pages 2357–2367, 2019.

Daniel Andor, Luheng He, Kenton Lee, and Emily Pitler. Giving BERT a calculator: Finding operations and arguments with reading comprehension. In Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan, editors, *EMNLP*, pages 5946–5951, 2019.

Ting-Rui Chiang and Yun-Nung Chen. Semantically-aligned equation generation for solving and reasoning math word problems. In *NAACL*, 2018.

Yiming Cui, Wanxiang Che, Ting Liu, Bing Qin, Shijin Wang, and Guoping Hu. Revisiting pre-trained models for chinese natural language processing. In *EMNLP: Findings*, pages 657–668, 2020.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL*, pages 4171–4186, 2019.

Dheeru Dua, Yizhong Wang, Pradeep Dasigi, Gabriel Stanovsky, Sameer Singh, and Matt Gardner. DROP: A reading comprehension benchmark requiring discrete reasoning over paragraphs. In Jill Burstein, Christy Doran, and Thamar Solorio, editors, *NAACL*, pages 2368–2378, 2019.

Mor Geva, Ankit Gupta, and Jonathan Berant. Injecting numerical reasoning skills into language models. In Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel R. Tetreault, editors, *ACL*, pages 946–958, 2020.

John Hewitt and Christopher D. Manning. A structural probe for finding syntax in word representations. In Jill Burstein, Christy Doran, and Thamar Solorio, editors, *NAACL*, pages 4129–4138, 2019.

Yining Hong, Qing Li, Daniel Ciao, Siyuan Huang, and Song-Chun Zhu. Learning by fixing: Solving math word problems with weak supervision. In *AAAI*, 2021a.

Yining Hong, Qing Li, Ran Gong, Daniel Ciao, Siyuan Huang, and Song-Chun Zhu. Smart: A situation model for algebra story problems via attributed grammar. In *AAAI*, 2021b.

Danqing Huang, Shuming Shi, Chin-Yew Lin, Jian Yin, and Wei-Ying Ma. How well do computers solve math word problems? large-scale dataset construction and evaluation. In *ACL*, pages 887–896, 2016.

Shifeng Huang, Jiawei Wang, Jiao Xu, Da Cao, and Ming Yang. Recall and learn: A memory-augmented solver for math word problems. In *Findings of EMNLP*, pages 786–796, 2021.

Zhanming Jie, Jierui Li, and Wei Lu. Learning to reason deductively: Math word problem solving as complex relation extraction. In *ACL*, pages 5944–5955, 2022.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Rik Koncel-Kedziorski, Hannaneh Hajishirzi, Ashish Sabharwal, Oren Etzioni, and Siena Dumas Ang. Parsing algebraic word problems into equations. *Transactions of the Association for Computational Linguistics*, 3:585–597, 2015.

Nate Kushman, Yoav Artzi, Luke Zettlemoyer, and Regina Barzilay. Learning to automatically solve algebra word problems. In *ACL*, pages 271–281, 2014.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. BART: denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel R. Tetreault, editors, *ACL*, pages 7871–7880, 2020.

Jierui Li, Lei Wang, Jipeng Zhang, Yan Wang, Bing Tian Dai, and Dongxiang Zhang. Modeling intra-relation in math word problems with different functional multi-head attentions. In *ACL*, pages 6162–6167, 2019.

Zhongli Li, Wenxuan Zhang, Chao Yan, Qingyu Zhou, Chao Li, Hongzhi Liu, and Yunbo Cao. Seeking patterns, not just memorizing procedures: Contrastive learning for solving math word problems. *arXiv preprint arXiv:2110.08464*, 2021.

Zhenwen Liang and Xiangliang Zhang. Solving math word problems with teacher supervision. In *IJCAI*, 2021.

Qianying Liu, Wenyv Guan, Sujian Li, and Daisuke Kawahara. Tree-structured decoding for solving math word problems. In *EMNLP*, pages 2370–2379, 2019.

Jinghui Qin, Lihui Lin, Xiaodan Liang, Rumin Zhang, and Liang Lin. Semantically-aligned universal tree-structured solver for math word problems. In Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu, editors, *EMNLP*, pages 3780–3789, 2020.

Jinghui Qin, Xiaodan Liang, Yining Hong, Jianheng Tang, and Liang Lin. Neural-symbolic solver for math word problems with auxiliary tasks. In *ACL*, pages 5870–5881, 2021.

Jianhao Shen, Yichun Yin, Lin Li, Lifeng Shang, Xin Jiang, Ming Zhang, and Qun Liu. Generate & rank: A multi-task framework for math word problems. In Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih, editors, *EMNLP*, pages 2269–2279, 2021.

Yibin Shen and Cheqing Jin. Solving math word problems with multi-encoders and multi-decoders. In *COLING*, pages 2924–2934, 2020.

Shuming Shi, Yuehui Wang, Chin-Yew Lin, Xiaojiang Liu, and Yong Rui. Automatically solving number word problems by semantic parsing and reasoning. In *EMNLP*, pages 1132–1142, 2015.

Minghuan Tan, Lei Wang, Lingxiao Jiang, and Jing Jiang. Investigating math word problems using pretrained multilingual language models. *CoRR*, abs/2105.08928, 2021.

Avijit Thawani, Jay Pujara, Filip Ilievski, and Pedro A. Szekely. Representing numbers in NLP: a survey and a vision. In Kristina Toutanova, Anna Rumshisky, Luke Zettlemoyer, Dilek Hakkani-Tür, Iz Beltagy, Steven Bethard, Ryan Cotterell, Tanmoy Chakraborty, and Yichao Zhou, editors, *NAACL-HLT*, pages 644–656, 2021.

Eric Wallace, Yizhong Wang, Sujian Li, Sameer Singh, and Matt Gardner. Do NLP models know numbers? probing numeracy in embeddings. In Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan, editors, *EMNLP*, pages 5306–5314, 2019.

Lei Wang, Yan Wang, Deng Cai, Dongxiang Zhang, and Xiaojiang Liu. Translating a math word problem to a expression tree. In *EMNLP*, pages 1064–1069, 2018.

Lei Wang, Dongxiang Zhang, Jipeng Zhang, Xing Xu, Lianli Gao, Bing Tian Dai, and Heng Tao Shen. Template-based math word problem solvers with recursive neural networks. In *AAAI*, volume 33, pages 7144–7151, 2019.

Yan Wang, Xiaojiang Liu, and Shuming Shi. Deep neural solver for math word problems. In *EMNLP*, pages 845–854, 2017.

Qinzhuo Wu, Qi Zhang, and Zhongyu Wei. An edge-enhanced hierarchical graph-to-tree network for math word problem solving. In *Findings of EMNLP*, pages 1473–1482, 2021a.

Qinzhuo Wu, Qi Zhang, Zhongyu Wei, and Xuanjing Huang. Math word problem solving with explicit numerical values. In *ACL*, pages 5859–5869, 2021b.

Zhipeng Xie and Shichao Sun. A goal-driven tree-structured neural model for math word problems. In *IJCAI*, pages 5299–5305, 2019.

Weijiang Yu, Yingpeng Wen, Fudan Zheng, and Nong Xiao. Improving math word problems with pre-trained knowledge and hierarchical reasoning. In *EMNLP*, pages 3384–3394, 2021.

Ma Yuhui, Zhou Ying, Cui Guangzuo, Ren Yun, and Huang Ronghuai. Frame-based calculus of solving arithmetic multi-step addition and subtraction word problems. In *2010 Second International Workshop on Education Technology and Computer Science*, volume 2, pages 476–479. IEEE, 2010.

Jipeng Zhang, Roy Ka-Wei Lee, Ee-Peng Lim, Wei Qin, Lei Wang, Jie Shao, and Qianru Sun. Teacher-student networks with multiple decoders for solving math word problem. In *IJCAI*, pages 4011–4017, 2020a.

Jipeng Zhang, Lei Wang, Roy Ka-Wei Lee, Yi Bin, Yan Wang, Jie Shao, and Ee-Peng Lim. Graph-to-tree learning for solving math word problems. In *ACL*, pages 3928–3937, 2020b.

Xikun Zhang, Deepak Ramachandran, Ian Tenney, Yanai Elazar, and Dan Roth. Do language embeddings capture scales? In Trevor Cohn, Yulan He, and Yang Liu, editors, *Findings of EMNLP*, volume EMNLP 2020, pages 4889–4896. Association for Computational Linguistics, 2020c.

Wei Zhao, Mingyue Shang, Yang Liu, Liang Wang, and Jingming Liu. Ape210k: A large-scale and template-rich dataset of math word problems. *arXiv preprint arXiv:2009.11506*, 2020.

Yanyan Zou and Wei Lu. Text2math: End-to-end parsing text into math expressions. In *EMNLP*, pages 5330–5340, 2019.

Appendix

Related Works

Math Word Problems Solving. There exist two major types of MWP, equation set MWP Wang et al. [2017], Zhao et al. [2020] and arithmetic MWP Qin et al. [2020], Huang et al. [2016]. This work focuses on arithmetic MWP, which is usually paired with one unknown variable. Along the path of the MWP solver's development, the pioneer studies use traditional rule-based methods, machine learning methods and statistical methods Yuhui et al. [2010], Kushman et al. [2014], Shi et al. [2015], Koncel-Kedziorski et al. [2015]. Afterward, inspired by the development of sequence-to-sequence (Seq2Seq) models, MWP solving has been formulated as a neurosymbolic reasoning pipeline of translating language descriptions to mathematical equations with encoder-decoder framework Wang et al. [2018, 2019], Li et al. [2019], Zhang et al. [2020b], Yu et al. [2021], Wu et al. [2021a]. By fusing hard constraints into decoder Chiang and Chen [2018], Liu et al. [2019], Xie and Sun [2019], Shen and Jin [2020], Zhang et al. [2020a], MWP solvers achieve much better performance then. Several works propose to utilize multi-stage frameworks Wang et al. [2019], Huang et al. [2021], Shen et al. [2021], Liang and Zhang [2021] to make more robust solvers. Also, several new works made attempts to improve MWP solver beyond supervised settings Hong et al. [2021a,b].

Among all these previous studies, the most relevant ones to our work can be categorized into two groups. First, it has been noted that number values and mathematical constraints play a significant role in supporting numerical reasoning. Wu et al. [2021b] proposed several number value features to enhance encoder and Qin et al. [2021] designed new auxiliary tasks to enhance neural MWP solvers. Compared with their work, we first introduce pre-training language model (PLM) and concentrate on representation learning to resolve numerical understanding challenges. Second, regarding the usage of pre-training techniques for MWP solving, Shen et al. [2021] introduced BART-based Lewis et al. [2020] MWP solver and incorporated specialized multi-task training for obtaining more effective pre-training Seq2Seq models for MWP. Compared with them, our work focuses on the number representation learning issue of MWP and achieves a more flexible pre-training representation module for MWP solving, which can be applied in various MWP-related tasks other than solution generation.

Numeracy-aware Pre-training Models. Number representation has been recognized as one of the main issues in word representation learning. Existing methods make use of value, exponent, sub-word and character methods Thawani et al. [2021] to obtain number representations for explicit number values. These methods are known to be less effective in extrapolation cases like testing with numbers not appearing in the training corpus.

Previous related works Andor et al. [2019], Wallace et al. [2019], Geva et al. [2020] mainly focus on shallow numerical reasoning tasks shown in DROP dataset Dua et al. [2019], which usually serves as

a benchmark for evaluating numerical machine reading comprehension (Num-MRC) performance. Compared with MWP solving, Num-MRC's main focus is laid on extracting answer spans from a paragraph, which are more fault-tolerant with no needs to predict number tokens. Besides, their solution generation tasks only contain simple computations like addition/subtraction and there are only integers in DROP. More exactly, several research efforts have been made to deal with this kind of math-related reading comprehension task by synthesizing new training examples Geva et al. [2020], incorporating special modules considering the numerical operation Andor et al. [2019] and designing specific tokenization strategies Zhang et al. [2020c]. Since MWP solving requires further consideration of the complex composition of reasoning logic in MWP text, the symbolic placeholder is more effective in MWP solving. Thus, instead of dealing with explicit number values, our work focuses on improving representation for symbolic placeholders by injecting numerical properties in a probabilistic way.

Implementation Details

We pre-train our model on 4 NVIDIA TESLA V100 graphic cards and fine-tune on 1 card. The model was pre-trained for 50 epochs (2 days) and fine-tuned for 80 epochs (1 day) with a batch size of 32. Adam optimizer Kingma and Ba [2014] is applied with an initial learning rate of 5e-5, which would be halved every 30 epochs. A dropout rate of 0.5 is set during training to prevent over-fitting. During testing, we use a 5-beam search to get reasonable solutions. The hyper-parameters setting of our BERT and RoBERTa is 12 layers of depth, 12 heads of attention and 768 dimensions of hidden features. Our code and data have been open-sourced on Github [3].

Ape-clean Dataset

Ape210k is a recently released large MWPs dataset Zhao et al. [2020], including 210,488 problems. The problems in Ape210k are more diverse and difficult than those in Math23k. Not only the stronger requirement of common-sense knowledge for getting solutions but also the missing of ground-truth solution equations or answers, will take extra obstacles for MWP solving. Among all these cases, the problems without answers can not be used in fully-supervised setting. Besides, the problems without annotated equations but only answer values can be used in the weakly-supervised learning setting. Therefore, we follow the rules below to select the usable problems from Ape210k to construct an Ape-clean dataset, which can be used for the fully-supervised learning setting. (i). We remove all MWPs that have no answer values nor equations. (ii). We remove all MWPs that only have answer values without equations. (iii). We remove all MWPs with a problem length $m > 100$ or an answer equation length $n > 20$, as they will bring obstacles for training. (iv). We remove all MWPs requiring external constants except 1 and π. (v). We remove all duplicated problems with the MWPs in Math23k because almost all problems in Math23k can be found in Ape-210k. After data filtering, the *Ape-clean* dataset contains 81,225 MWPs, including 79,388 training problems and 1,837 testing problems. The remaining 129,263 problems in Ape210k are regarded as *Ape-unsolvable*, which can be used in the pre-training tasks in the settings of self-supervised and weakly-supervised learning.

[3] `https://github.com/LZhenwen/MWP-BERT`