Noun-MWP: Math Word Problems Meet Noun Answers

Taehun Cha* and **Jaeheun Jung*** and **Donghun Lee**
Department of Mathematics
Korea University
{cth127,wodsos,holy}@korea.ac.kr

Abstract

We introduce a new type of problems for math word problem (MWP) solvers, named Noun-MWPs, whose answer is a non-numerical string containing a noun from the problem text. We present a novel method to empower existing MWP solvers to handle Noun-MWPs, and apply the method on Expression-Pointer Transformer (EPT). Our model, N-EPT, solves Noun-MWPs significantly better than other models, and at the same time, solves conventional MWPs as well. Solving Noun-MWPs may lead to bridging MWP solvers and traditional question-answering NLP models.

1 Introduction

Question-answering (QA) (Woods, 1968) and QA problems involving mathematics (Bobrow, 1964) are one of the classic NLP problems in computer science. Later, mathematics education community investigated a class of problems coined as Math Word Problems (MWPs) (Nesher and Katriel, 1977; Caldwell and Goldin, 1979; Ballew and Cunningham, 1982). Research in solving MWPs with computers started from purely rule-based approaches (Hosseini et al., 2014) and statistics-based approaches (Kushman et al., 2014), through hybrids (Roy and Roth, 2015), and more recently deep-learning-based approaches (Wang et al., 2017, 2018; Li et al., 2019).

Proportionally, the number of publicly available dataset for MWPs increased; for example, `Alg514` (Kushman et al., 2014), `DRAW` (Upadhyay and Chang, 2015), `MAWPS` (Koncel-Kedziorski et al., 2016), `Math23K` (Wang et al., 2017), `AllArith` (Roy and Roth, 2017), `Dolphin18K` (Huang et al., 2016), and `ASDiv-A` (Miao et al., 2020). The mathematical depth of the problems in MWP datasets vary, yet the majority of datasets require only numerical answers. On the other hand, math problems whose answer is a non-numeric substring of the inputs are frequently seen in real-world math problems in mathematics education.

MWP
Chloe was organizing her bookcase making sure each of the shelves had exactly 6 books on it. If she had 5 shelves of mystery books and 4 shelves of picture books, how many books did she have in total?
$\rightarrow (6 \times 5) + (6 \times 4)$
Extractive QA
In 1517, the seventeen-year-old King sailed to Castile. There, his Flemish court . . . . In May 1518, Charles traveled to Barcelona in Aragon. Where did Charles travel to first, Castile or Barcelona?
$\rightarrow arg\ min\{Castile : 1517, Barcelona : 1518\}$
Noun-MWP
Sihyeon collected 64 stamps, and Soma collected 1 fewer stamps than Sihyeon. Junwoo collected 7 bundles of 10 stamps each. Who among the three has collected the fewest stamps?
$\rightarrow arg\ min\{Sihyeon : 64, Soma : 64 - 1, Junwoo : 7 \times 10\}$

Table 1: Highlighted examples comparing MWP, Noun-MWP, and Extractive QA problems. Relevant numbers, nouns, and a novel element in Noun-MWP are marked in red, blue, and green, respectively.

Our Contribution

- We bring attention to Noun-MWP, a class of MWPs whose answer is a noun-substring of the input string.
- We propose a systematic approach to empower existing MWP solvers to solve Noun-MWPs as well.
- We implement and empirically validate our approach using a Korean Noun-MWP dataset.

*equal contribution

Proceedings of the 29th International Conference on Computational Linguistics, pages 3847–3857
October 12–17, 2022.

2 Related Works

Our work emphasizes the need for a new class of MWPs to complement the MWPs represented by existing datasets. From a similar stance, ASDiv-A dataset (Miao et al., 2020) emphasizes the need to have greater lexical diversity in MWP datasets, and S-VAMP dataset (Patel et al., 2021) emphasizes the underrepresented adversarial variations of the MWPs. Noun-MWP may be considered as a next-level adversarial variations of pre-existing MWPs, since as of now, adversarial generation methods of MWPs (Kumar et al., 2021) do not exceed the current implicit limit of MWPs focused on problems with numerical answers.

Noun-MWPs can also be seen as extractive QA problems that need mathematics-based reasoning to reach the correct solution. DROP dataset (Dua et al., 2019), as shown in the middle panel of Table 1, introduces basic mathematical reasoning in QA tasks. The direct approach to populate mathematical operations with arguments (Andor et al., 2019) faces limitation due to combinatorial explosion of search space in possible mathematical expressions, and injecting custom math operations and their templates (Geva et al., 2020) alleviate this drawback somewhat. To circumvent this, we start from MWP perspective and use Noun-MWPs as a gateway to solve extractive QA problems whose solutions contain more complicated mathematical expressions.

3 Methodology

3.1 Problem Definition

We consider Noun-MWP as a special type of math word problems whose correct answer is a string containing a noun from the problem context. In particular, we consider problems requiring algebraic operations $+, \times, -, \div$ and comparison between numbers. The difficulty of these problems roughly corresponds to the problems taught in elementary school level mathematics classes. Table 1 and Table 2 contain two examples.

To infer a correct solution from the problem in Table 2, a well-constructed series of mathematical inference and language processing is needed. First of all, how much milk Eunhye drank must be inferred from '3/7' and 'rest.' Then, the inferred number '4/7' must be assigned to the corresponding noun 'Eunhye.' Finally, based on the query, the correct noun answer must be determined. For an NLP model to solve a collection of noun-MWPs, both MWP and extractive QA capabilities are needed, especially in handling multi-step computation comprised of multiple mathematical operations correctly.

Problem(KR)
아린이는 우유 한 병 전체의 3/7을 마시고, 나머지는 은혜가 마셨습니다. 둘 중 우유를 더 적게 마신 사람은 누구일까요?
Problem(EN)
Arin drank 3/7 of the whole bottle of milk, and Eunhye drank the rest. Which of the two drank less milk?
Candidates : Arin, Eunhye Expressions : {Arin : $3/7$, Eunhye : $1 - 3/7$} Query : Which of the two drank less milk?

Table 2: A Noun-MWP example question presented in original Korean language and its translation to English, and the sample solution comprised of candidates (in blue), mathematical expressions, and query.

3.2 Solving Noun-MWP

In this section, we sketch the key idea of solving Noun-MWPs starting with a MWP solver. We decompose the process of solving Noun-MWP into three conceptual steps:

Candidate selection: Extract the candidate nouns from the context and generate the list of candidates, as shown in Table 2.

Expression assignment: Generate mathematical expressions that produce numbers. These numbers may be assigned to the candidates.

Query interpretation: Interpret the query into a representation compatible to the underlying MWP solver. Resulting representation allows the MWP solver to find the most suitable candidate to answer.

By implementing the three-step process on a multi-equation MWP solver, it is possible to empower the solver with necessary information to solve Noun-MWPs while preserving its original capacity to solve MWPs.

3.3 N-EPT: Noun problem solving EPT

To demonstrate the validity of our approach, we use EPT (Kim et al., 2020) as the baseline multi-equation MWP solver and empower it to solve Noun-MWPs. The original EPT solves the MWPs with the binary tree-based expression tokens, which consist of an operation and operands. They used

TAR(target) Contrast to special token VAR(variable) of original EPT, we treat it as an unary operator paired with target noun entity appeared in the problem. This allows the model to infer the candidates and equations to assign values to the noun entities.
arg Special command to generate list of pairs (noun, value) after the expression assignment. This works to separate classic MWPs, which don't require noun information, from our problem.
find and **ord(order)** Binary list operators extracting answer from the list(resp. sorted list) satisfying query. For example, (ord,R_i, n) means the n-th smallest element of the list stored at R_i. Similarly the (find,R_i, n) works on R_i and choose element indexed by n.

Table 3: Details on new operations of N-EPT, in constrast to EPT.

$+, -, \times, \div$ as operations, and applied an attention block to extract operands from the problem text. Maintaining the backbone structure of the EPT, our resulting MWP solver, named N-EPT[1], contains the following key changes:

- **Additional Operation Tokens**: We introduce four more operations, *TAR, arg, find, ord* to extend the expressive power of EPT to cover Noun-MWPs. Tokens and their operations are detailed in Table 3.
- **Noun attention**: An attention module is added to predict candidate nouns from the question text.
- **Pretrained word embedding**: N-EPT uses pretrained T5 embedding. For example, a pretrained embedding of a word *subtract* is used to represent subtracting operation.
- **T5 encoder-decoder**: In contrast to EPT, which uses ALBERT encoder and transformer decoder, N-EPT uses pretrained T5 encoder and decoder (KETI AIRC, 2021). This allows utilizing pretrained word embeddings and language-specific pretrained models.

[1]The code is available on https://github.com/invigorator96/NounMWP

Additional procedures were added to handle agglutinative characteristic of Korean language – for example, postpositions included in predicted nouns are omitted to facilitate exact answer matching. An implementation example is in Appendix A.

4 Experimental Study

4.1 Dataset

We construct a novel dataset comprised of Noun-MWPs for empirical validation. We collect Noun-MWP problems and their answers from elementary school level mathematics textbooks (최순미, 2020; 한헌조, 2018; 김은영, 2018) in South Korea, and label their mathematical expression manually. In addition, problems from a Korean MWP dataset curated by TUNiB (Keum et al., 2022) are also selected and relabeled. We validated the annotated solution labels against the answers from the source textbooks, by checking whether a rule based solver provided with the solution label can produce the same answer as the answers from the source textbooks. The resulting dataset contains 604 question-expression-answer triplets,whose basic statistics are in Table 4.

Ratio of Problems by Required Operations				
$+$	$-$	$\times$	$\div$	Simple Assignment
20.9%	8.4%	18.4%	9.3%	55.8%

Ratio of Problems by Expression Length				
≤ 8	$9 \sim 10$	$11 \sim 12$	$13 \sim 14$	≥ 15
46.5%	39.4%	9.6%	4.1%	0.3%

Table 4: Statistics of our Noun-MWP dataset

Simple Assignment cases in Table 4 correspond to the problems solved with only simple comparison or sorting, without any arithmetic operation. Our dataset includes problems requiring more than two operations to solve, so the percentages do not sum up to 100. The expression length of a problem represents the complexity of mathematical reasoning to deduce the correct answer, and is an intuitive gauge of problem difficulty.

4.2 Noun-MWP Performance

Using the Korean Noun-MWP dataset, we validate the performance of N-EPT implementation

via 5-fold cross validation. As shown in Table 5, N-EPT achieves over 80% in cross validated accuracy measured via exact match score, regardless of the choice of T5 encoder-decoder size – small, base, or large. Detailed methods used in training N-EPT and in-depth performance results are in Appendix B.

	N-EPT	KE-T5	KLUE-RoBERTa
Small	81.67%	−%	45.67%
(Std.Err)	(2.11%)	(-%)	(4.52%)
Base	**84.33%**	6.80%	46.17%
(Std.Err)	(2.44%)	(9.58%)	(5.21%)
Large	82.50%	28.17%	47.67%
(Std.Err)	(0.75%)	(2.44%)	(3.22%)

Table 5: 5-fold CV accuracy in Noun-MWPs, by N-EPT and two SOTA QA models.

Conventional MWP solvers cannot handle Noun-MWPs, as those solvers generate numerical answers. As an alternative, we use state-of-the-art (SOTA) QA models as benchmark algorithms, since QA models are capable of produce non-numerical answers based on the given question.

The first benchmark QA model is KE-T5, a SOTA Korean-pretrained version of T5 (Raffel et al., 2020) that achieves 86.27% accuracy on KorQuAD 1.1 dataset (Korean extractive QA benchmark) (KETI AIRC, 2021). KE-T5 fails to answer a great majority of Noun-MWP questions; whereas N-EPT with the same model specification consistently performs much better. As our N-EPT implementation contains KE-T5, the difference in performance between the two suggests that the expressive power added to KE-T5 by changes in Section 3.3 is very effective in solving Noun-MWPs.

We also use KLUE-RoBERTa (Park et al., 2021), a SOTA Korean-pretrained variant of RoBERTa (Liu et al., 2019) that achieves 75.58% accuracy on KLUE-MRC (Korean machine reading comprehension) benchmark as another benchmark QA model. KLUE-RoBERTa may be handling a smaller subset of problems in our Noun-MWP dataset than N-EPT, as it shows consistent performance that peaks at 47.67%, much less than 81.67%, the worst performance of N-EPT. Significant empirical advantage of N-EPT over both QA models suggest that empowering a conventional MWP solver is an effective approach to solve Noun-MWPs.

4.3 Sanity Check with Conventional MWPs

We backtest N-EPT on `MAWPS` dataset to verify that our approach retains the capacity of the original MWP solver (EPT) to handle conventional MWPs. Since `MAWPS` dataset is in English, we modify N-EPT structure by substituting KE-T5 with Google T5. Meanwhile, as the original implementation of EPT uses ALBERT, we also use our own implementation of EPT with T5 as another benchmark. We report the mean and the standard deviation of 5-fold cross-validated accuracy in Table 6.

	ALBERT model size		
	base	large	xlarge
EPT(orig.)	83.41%	84.51%	−%
(Std.Err)	(0.32%)	(1.37%)	(-%)
	T5 model size		
	small	base	large
EPT(T5)	83.35%	84.03%	85.12%
(Std.Err)	(1.70%)	(1.80%)	(1.92%)
N-EPT	83.27%	84.49%	85.50%
(Std.Err)	(1.85%)	(1.59%)	(1.53%)

Table 6: 5-fold CV accuracy in MAWPS dataset. EPT(orig.) values are from the original source (Kim et al., 2020).

Similar performance between EPT with ALBERT and EPT with T5 suggests that the choice of the encoder-decoder model has negligible effect on performance. Meanwhile, unlike ALBERT that shows instabilities in a larger model specification, T5 is stable throughout all specifications.

We implement EPT with T5 by ablating N-EPT, such that it does not have the other changes shown in Section 3.3 to empower EPT for Noun-MWPs besides adding T5. Therefore, the similar performance of the two suggests that N-EPT successfully retains the expressive power of EPT in handling conventional MWPs.

5 Discussion and Future Work

N-EPT demonstrates how to empower a conventional MWP solver to handle Noun-MWPs. Considering Noun-MWPs are a staple of basic math education, they deserve to be included in the benchmark for a novel MWP solver to gauge its ability to understand both the mathematics and the natural language in the problems.

Despite the ability to handle both conventional

MWPs and Noun-MWPs, N-EPT leaves much to be improved against harder questions. For example, problems whose expression length is longer or whose question string contains more candidates are often incorrectly answered by N-EPT. Detailed plots are placed in Appendix C, and select examples where N-EPT incorrectly answers are given in Appendix D.

6 Conclusion

We introduce Noun-MWP, a class of problems that are frequently used in mathematics education, and yet to be handled by MWP solvers. We present a three-step method, Candidate Selection, Expression Assignment, and Query Interpretation, to empower existing MWP solvers to handle Noun-MWP as well as the conventional MWPs. As a proof-of-concept implementation, we construct N-EPT using EPT as its baseline MWP solver, and the new Noun-MWP dataset consists of 604 questions-expression-answer pairs to validate the model's performance. By modifying the original EPT to handle noun information and additional operations, our N-EPT significantly outperforms benchmark models in Noun-MWPs and retains the capacity of the original MWP solver. Noun-MWPs may serve as a novel testbed for MWP solvers to gauge their understanding of mathematical logic and natural language.

Acknowledgements

This work is partly supported by Institute for Information & communications Technology Promotion (IITP) grant funded by the Korea government (MSIT) (No. 2021-0-02153, Large scale deep learning based algorithm for mathematics problem solving), and by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No. 2020R1G1A1102828). We thank Hanyoung Kim, Dongwon Kim, Hansol Jeon, Daeseong Kim, Yanggee Kim, Hyunsang Hwang, Keunsuk Cho, and Changhae Jung for their assistance in data collection and processing.

References

Daniel Andor, Luheng He, Kenton Lee, and Emily Pitler. 2019. Giving BERT a calculator: Finding operations and arguments with reading comprehension. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5947–5952, Hong Kong, China. Association for Computational Linguistics.

Hunter Ballew and James W Cunningham. 1982. Diagnosing strengths and weaknesses of sixth-grade students in solving word problems. *Journal for research in mathematics education*, 13(3):202–210.

Daniel G Bobrow. 1964. A question-answering system for high school algebra word problems. In *Proceedings of the October 27-29, 1964, fall joint computer conference, part I*, AFIPS '64 (Fall, part I), pages 591–614, New York, NY, USA. Association for Computing Machinery.

Janet H Caldwell and Gerald A Goldin. 1979. Variables affecting word problem difficulty in elementary school mathematics. *Journal for Research in Mathematics Education*, 10(5):323–336.

Dheeru Dua, Yizhong Wang, Pradeep Dasigi, Gabriel Stanovsky, Sameer Singh, and Matt Gardner. 2019. DROP: A reading comprehension benchmark requiring discrete reasoning over paragraphs. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2368–2378, Minneapolis, Minnesota. Association for Computational Linguistics.

Mor Geva, Ankit Gupta, and Jonathan Berant. 2020. Injecting numerical reasoning skills into language models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 946–958, Online. Association for Computational Linguistics.

Mohammad Javad Hosseini, Hannaneh Hajishirzi, Oren Etzioni, and Nate Kushman. 2014. Learning to solve arithmetic word problems with verb categorization. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 523–533, Doha, Qatar. Association for Computational Linguistics.

Danqing Huang, Shuming Shi, Chin-Yew Lin, Jian Yin, and Wei-Ying Ma. 2016. How well do Computers Solve Math Word Problems? Large-Scale Dataset Construction and Evaluation. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 887–896, Berlin, Germany. Association for Computational Linguistics.

KETI AIRC. 2021. KE-T5: Korean English T5. https://github.com/AIRC-KETI/ke-t5.

Bitna Keum, Myeonghyeon Ryu, Yoseph Ham, Minsuh Seo, Heechang Jo, Hangyeol Kim, and Kyubyong Park. 2022. Kmwp, korean math word problems. https://github.com/tunib-ai/KMWP.

Bugeun Kim, Kyung Seo Ki, Donggeon Lee, and Gahgene Gweon. 2020. Point to the Expression: Solving Algebraic Word Problems using the Expression-Pointer Transformer Model. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3768–3779.

Rik Koncel-Kedziorski, Subhro Roy, Aida Amini, Nate Kushman, and Hannaneh Hajishirzi. 2016. MAWPS: A Math Word Problem Repository. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1152–1157, San Diego, California. Association for Computational Linguistics.

Vivek Kumar, Rishabh Maheshwary, and Vikram Pudi. 2021. Adversarial examples for evaluating math word problem solvers. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 2705–2712, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Nate Kushman, Yoav Artzi, Luke Zettlemoyer, and Regina Barzilay. 2014. Learning to Automatically Solve Algebra Word Problems. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 271–281, Baltimore, Maryland. Association for Computational Linguistics.

Jierui Li, Lei Wang, Jipeng Zhang, Yan Wang, Bing Tian Dai, and Dongxiang Zhang. 2019. Modeling Intra-Relation in Math Word Problems with Different Functional Multi-Head Attentions. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6162–6167, Florence, Italy. Association for Computational Linguistics.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A Robustly Optimized BERT Pretraining Approach.

Shen-yun Miao, Chao-Chun Liang, and Keh-Yih Su. 2020. A diverse corpus for evaluating and developing English math word problem solvers. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 975–984, Online. Association for Computational Linguistics.

Pearla Nesher and Tamar Katriel. 1977. A Semantic Analysis of Addition and Subtraction Word Problems in Arithmetic. *Educational Studies in Mathematics*, 8(3):251–269.

Sungjoon Park, Jihyung Moon, Sungdong Kim, Won Ik Cho, Ji Yoon Han, Jangwon Park, Chisung Song, Junseong Kim, Youngsook Song, Taehwan Oh, Joohong Lee, Juhyun Oh, Sungwon Lyu, Younghoon Jeong, Inkwon Lee, Sangwoo Seo, Dongjun Lee, Hyunwoo Kim, Myeonghwa Lee, Seongbo Jang, Seungwon Do, Sunkyoung Kim, Kyungtae Lim, Jongwon Lee, Kyumin Park, Jamin Shin, Seonghyun Kim, Lucy Park, Lucy Park, Alice Oh, Jung-Woo Ha (NAVER AI Lab), Kyunghyun Cho, and Kyunghyun Cho. 2021. KLUE: Korean Language Understanding Evaluation. In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*, volume 1.

Arkil Patel, Satwik Bhattamishra, and Navin Goyal. 2021. Are NLP models really able to solve simple math word problems? In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2080–2094, Online. Association for Computational Linguistics.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67.

Subhro Roy and Dan Roth. 2015. Solving general arithmetic word problems. In *Conference on Empirical Methods in Natural Language Processing, EMNLP 2015*, pages 1743–1752. experts.illinois.edu.

Subhro Roy and Dan Roth. 2017. Unit Dependency Graph and Its Application to Arithmetic Word Problem Solving. *Proceedings of the AAAI Conference on Artificial Intelligence*, 31(1).

Shyam Upadhyay and Ming-Wei Chang. 2015. Draw: A challenging and diverse algebra word problem set. Technical report, Citeseer.

Lei Wang, Yan Wang, Deng Cai, Dongxiang Zhang, and Xiaojiang Liu. 2018. Translating a math word problem to a expression tree. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1064–1069, Brussels, Belgium. Association for Computational Linguistics.

Yan Wang, Xiaojiang Liu, and Shuming Shi. 2017. Deep neural solver for math word problems. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 845–854, Copenhagen, Denmark. Association for Computational Linguistics.

W A Woods. 1968. Procedural semantics for a question-answering machine. In *Proceedings of the December 9-11, 1968, fall joint computer conference, part I*, AFIPS '68 (Fall, part I), pages 457–471, New York, NY, USA. Association for Computing Machinery.

Qinzhuo Wu, Qi Zhang, Jinlan Fu, and Xuanjing Huang. 2020. A knowledge-aware sequence-to-tree network for math word problem solving. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7137–7146, Online. Association for Computational Linguistics.

김은영. 2018. *기적의 수학 문장제*. 길벗스쿨, Seoul,KR.

징검다리 교육연구소 최순미. 2020. *나 혼자 푼다! 수학 문장제 - 서술형 기본서 :초등*. 이지스에듀, Seoul,KR.

김성국 한헌조. 2018. *씨투엠수학독해*. (주)씨투엠에듀, Suwon,KR.

Appendix

A Implementation Example

The sequence of expressions in Table A1 shows how N-EPT uses its operations to solve the sample Noun-MWP question shown in Table 2. Note that

Label	Expression	
R_0	$[TAR, Arin]$	Candidate selection
R_1	$[TAR, Eunhye]$	
R_2	$[=, R_0, 3/7]$	Expression assignment
R_3	$[-, 1, R_0]$	
R_4	$[=, R_1, R_3]$	
R_5	$[arg]$	Query interpretation
R_6	$[ord, R_5, 1]$	
R_7	$[END]$	

Table A1: N-EPT expression sequence to solve the Noun-MWP in Table 2.

R_n stands for the n-th token. The solver first select candidates with R_0 and R_1, and assign proper expressions to each candidates with $R_2 \sim R_4$. Then a special command R_5 construct a list of (candidate, expression) pairs, and finally, R_6 determine the query.

B Experiment Details

Five A6000 GPUs are utilized to run all folds parallelly, and we checked accuracy score on test set for every 20 training epochs. We used all different batch sizes for each model but used gradient accumulation technique to update gradients of 64 problems uniformly. After hyperparameter tuning with learning rate from [3e-4, 5e-5, 1e-5] with an ADAMW optimizer and dropout probability from [0.0, 0.1], 3e-4 and 0.1 were optimal for each. We also used label smoothing, gradient clipping and warm up technique as an original EPT did.

As a result, N-EPT achieved following results.

500epoch column presents final accuracy of all folds while *max* column shows the best accuracy on each fold. Though the *Base* model achieved the best performance among others, the *Large* model presented the smallest standard deviation on both columns. It can be an evidence of robustness of the *Large* model.

	N-EPT	
	500epoch	max
Small	79.67 ±2.33%	81.67 ±2.11%
Base	84.00 ±2.44%	**84.33** ±2.44%
Large	81.00 ±0.97%	82.50 ±0.75%

Table B2: 5-fold cross validated performance of new algorithm on Noun-MWPs.

C Impact of question complexity

We checked the impact of expression length by visualizing error rate and error count for each length. We used *base* model and gathered test sets in all five folds. It is natural to assume that error rate grows proportionally to expression length, because a question with longer expression is harder to solve. Figure C1 below shows consistent result with our assumption(Orange bar shows error rates for each category).

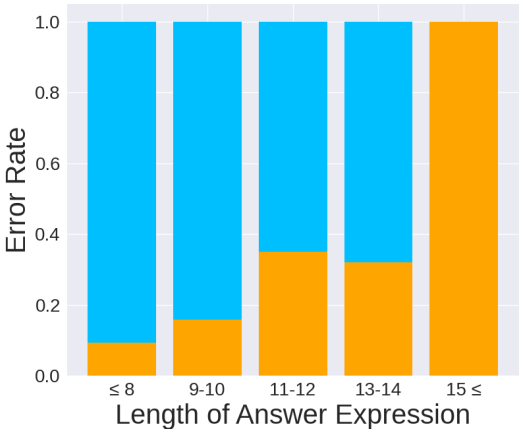

Figure C1: Impact of expression length to error rate.

There are two reasons to explain this result. The first is the relationship between expression length and question complexity. It is consistent with the report on (Wu et al., 2020), which showed decreasing performance of various models with respect to the equation length on Math23K dataset. The second is the sparsity of data with long expression in our dataset. Recall the Table 4 showing that problems in each category is highly skewed to short problems. Similar phenomena is observed when we visualize the error rate with the number of targets, candidates for answer.

Error rate is low with five targets because easy question requiring a 'find' operator consists most

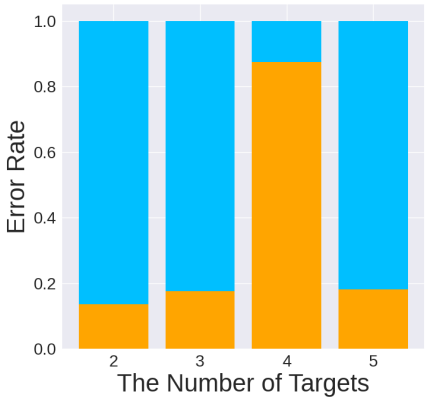

Figure C2: Impact of the number of TAR to error rate.

of the category. When there's too many targets to catch, model tends to miss some of them. Examples are presented in Appendix D. Since the distribution of the number of target is also highly skewed, same explanations mentioned above can be applied here, too.

To check the absolute count of error case and the number of problems of each category, see Figure C3 and C4.

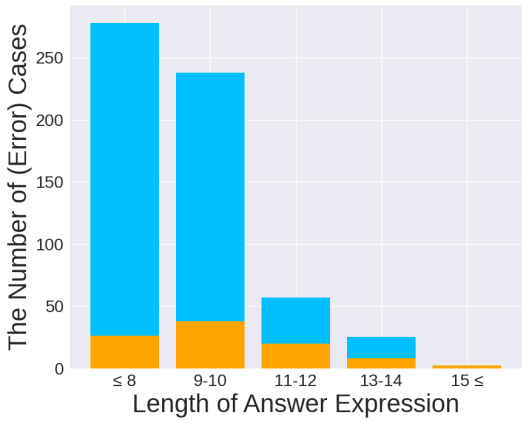

Figure C3: Impact of expression length to the number of error cases.

D Error Analysis

In this section, we make qualitative analysis on error cases. We implemented same procedure from Appendix C. Our model showed worse performance on complicated question requiring longer expressions. The Table D3 shows an example.

Though our model succeeded to find candidate nouns (*Sujin, Cheolmin* and *Youngsoo*), catch the intention of the query(*the most*) and compute the

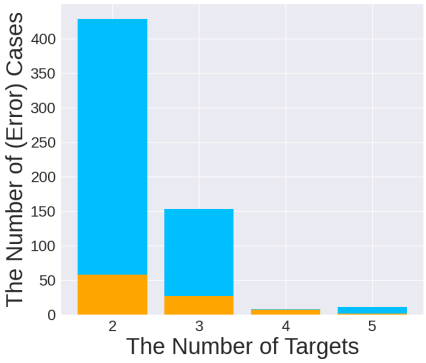

Figure C4: Impact of the number of TAR to the number of error cases.

Problem(EN)
Sujin drank 300mL of milk for two days, and Cheolmin drank 1,000mL more than three times the milk Sujin drank. Young-soo drank 1,000mL of milk every day for 12 days. Who drank the most milk?
Predicted expression
X_0: (TAR, Sujin), X_1: (TAR, Cheolmin), X_2: (TAR, Youngsoo) 300×2, $X_0 = X_2$, $X_0 \times 12$, $X_1 \times 12$ (arg), (ord, R_i, -1)

Table D3: Error case of our model

X_0 value(300×2), it failed to assign it to X_0 and successively failed to generate proper expressions. This shows the difficulty of generating long sequences, which generative models usually goes through.

	count	percent
Candidate Selection	55	58.5%
Expression Assignment	22	23.4%
Query Interpretation	17	18.1%
Total	94	100.0%

Table D4: Counts and percentage of error cases for each step. We added the number of error cases for all five folds.

In addition to the length issue, various reasons made error cases. As shown in Table D4, more than half of the error cases came from the Candidate Selection step. To improve the model, it would be necessary to enhance the Candidate Selection step, which is left to future research. We conclude the section with sample problems that our model failed to give a correct answer. The selected problems are sampled from various test folds in 5-fold cross validation experiment.

Problem(KR)
국제 어린이 학교에 중국 학생이 53명, 미국 학생은 60명 있습니다. 한 모둠에 10명씩 모았을 때 일본 학생은 5모둠에 5명이 남고, 한국 학생은 4모둠에 9명이 남습니다. 학생이 두 번째로 많은 나라는 어디인지 써 보세요.

Problem(EN)
There are 53 Chinese students and 60 American students in international children's schools. When 10 students are gathered in a group, 5 Japanese students are left in 5 groups, and 9 Korean students are left in 4 groups. Write down which country has the second largest number of students.

True expressions	
Candidate selection:	X_0:Chinese, X_1: American, X_2: Japanese, X_3: Korean
expression assignment:	$X_0 = 53$, $X_1 = 60$, $X_2 = 5 \times 10 + 5$, $X_3 = 4 \times 10 + 9$
Query interpretation:	(arg),(ord,R_i,-2)
Answer:	Japanese
Predicted expressions	
Candidate selection:	X_0:Chinese, X_1: American, X_2: Korean
expression assignment:	$X_0 = 53$, $X_1 = 60$, $X_2 = 9$
Query interpretation:	(arg),(ord,R_i,2)
Answer:	Chinese

Table D5: Incorrect sample problem. The model failed to find candidate **Japanese**, successively the expressions and query. Note that this is the problem with longest expression with expression length 15.

Problem(KR)
학교에서 국어, 수학, 영어, 과학, 사회의 순서로 시험을 봤습니다. 두 번째로 시험을 본 과목은 무엇입니까?

Problem(EN)
At school, I took the test in the order of Korean, math, English, science, and social studies. What subject did I take the exam for the second time?

True expressions	
Candidate selection:	X_0:Korean, X_1: math, X_2: English, X_3: Science, X_4:Social studies
expression assignment:	$X_0 = 1$, $X_1 = 2$, $X_2 = 3$,$X_3 = 4$,$X_4 = 5$
Query interpretation:	(arg),(find,R_i,2)
Answer:	math
Predicted expressions	
Candidate selection:	X_0:Korean, X_1: math, X_2: English, X_3: Science, X_4:Social studies
expression assignment:	$X_0 = 1$, $X_1 = 2$, $X_2 = 3$,$X_3 = 4$,$X_4 = 5$
Query interpretation:	(arg),(find,R_i,3)
Answer:	English

Table D6: Incorrect sample problem. The model successfully found candidates and intended expressions, but failed to find index operand of **find** operator.

Problem(KR) 선희는 30분에 10/3km를 걷고, 진혜는 30분에 2와 2/3km를 걷습니다. 더 천천히 걷는 사람은 누구일까요?
Problem(EN) Sun-hee walks 10/3 kilometers in 30 minutes, and Jin-hye walks 2 and 2/3 kilometers in 30 minutes. Who walks more slowly?
True expressions
Candidate selection: X_0:Sun-hee, X_1: Jin-hye
expression assignment: $X_0 = (10/3)$,$X_1 = (2 + (2/3))$
Query interpretation: (arg),(ord,R_i,1)
Answer: Jin-hye
Predicted expressions
Candidate selection: X_0:Sun-hee, X_1: Jin-hye
expression assignment: $X_0 = (10/3) \div 30$, $X_1 = 2 + (2/3)$
Query interpretation: (arg),(ord,R_i,1)
Answer: Sun-hee

Table D7: Incorrect sample problem. The model successfully found candidates and question token, but not correct expressions. The model tried to find the velocity of Sun-hee but not for Jin-hye.

Problem(KR) A는 한 변의 길이가 5cm인 정사각형, B는 가로가 5cm, 세로가 8cm인 직사각형, C는 한 변의 길이가 3cm인 정칠각형입니다. 이중에서 , 둘레가 가장 긴 도형을 찾으세요.
Problem(EN) A is a square with a side length of 5 cm, B is a rectangle with a width of 5 cm and a height of 8 cm, and C is a regular heptagon with a side length of 3 cm. Find the shape with the longest circumference.
True expressions
Candidate selection: X_0: A, X_1: B, X_2: C
expression assignment: $X_0 = 5 \times 4$, $X_1 = 2 \times (5 + 8)$, $X_2 = 3 \times 7$
Query interpretation: (arg),(ord,R_i, -1)
Answer: B
Predicted expressions
Candidate selection: X_0: A, X_1: B, X_2: C
expression assignment: $X_0 = 5 \times 10$, $X_1 = 2 \times (5 + 8)$, $X_2 = 3 \times 5$
Query interpretation: (arg),(ord,R_i, -1)
Answer: A

Table D8: Incorrect sample problem. The prior knowledge of square,rectangle and regular heptagon is required to solve this problem, but our model did not learn it. As a consequence, the expression assignment is not correct. Note that the model succeeded to get circumference of rectangle.