

Does ChatGPT Comprehend Place Value in Numbers When Solving Math Word Problems?*

Jisu An^{1,*}, Junseok Lee¹ and Gahgene Gweon¹

¹Seoul National University, South Korea

Abstract

With the recent advancements in GPT, there's been a growing trend to integrate GPT in solving math word problems using strategies such as "Chain-of-Thought"(CoT) and "Program-of-Thought"(PoT). Based on the observation that CoT tends to yield lower accuracy than PoT when large numbers are involved, we conducted two experiments to examine whether chatGPT understands place values in numbers. In the first experiment, to examine whether GPT can correctly order numbers based on an understanding of place values, we order source and permutation multiplets that contain 3-6 numbers in base 1000. In the second experiment, We examine whether GPT based models that utilize English expressions (CoT_Eng and PoT_Eng), rather than numerical expressions (CoT_Num and PoT_Num) can yield better performance in solving math word problems. The results of the first experiment showed that the ordering accuracy for the permutation multiplets (6 elements = 60.5%) was lower than that of source multiplets (6 elements = 96.8%). The results of the second experiment showed that accuracy increased when information about the place value was provided explicitly in the format of English expression (79.9% in CoT, 82% in PoT) compared to numerical expression(76.8% in CoT, 80% in PoT). The observations from both experiments suggest that the concept of place value isn't adequately integrated when numbers are represented as tokens in gpt3.5-turbo. Thus, research on training models to understand the concept of place value in numbers would be a possible direction to pursue as future research.

Keywords

Math word problem, place value, ChatGPT

1. Introduction

Recent advances in AI, such as chatGPT, have led to a surge in studies investigating how AI can be utilized to assist learning in education [1]. In particular, AI-augmented math word problem-solving has been attempted by researchers using variants of chatGPT, including chain of thought(CoT) [2] and programming of thought(PoT) [3]. Math word problems involve deriving equations or answers using the information given in a scenario that reflects a common daily-life situation. Although math word problems have been extensively used in mathematical education, these problems are also among the most challenging for math learners [4]. Given the potential of chatGPT in solving math word problems, we explore its number comprehension ability in terms of understanding place value. Since both words and numbers are treated equally

AIED 2023 Workshop: Towards the Future of AI-Augmented Human Tutoring in Math Learning, July 07, 2023, Tokyo, Japan


*Corresponding author.

✉ ajs7270@snu.ac.kr (J. An); shawn159@snu.ac.kr (J. Lee); ggweon@snu.ac.kr (G. Gweon)

🆔 0009-0006-5235-3408 (J. An); 0009-0000-2482-052X (J. Lee); 0000-0003-3268-477X (G. Gweon)



© 2023 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

 CEUR Workshop Proceedings (CEUR-WS.org)

as tokens when the chatGPT model is being trained, we suspect that the model may not have learned the concept of place value, which is a unique but necessary concept in understanding numbers. In particular, given that recent attempts at using chatGPT to solve math word problems have shown potential for using the model in education, understanding how chatGPT comprehends numbers would provide a better insight into the extent to which the model can be used in math education.

Given that chatGPT can understand and respond to inputs in natural language, making it feel intuitive and natural for users, academic research on education using chatGPT is in active progress [5]. Several prompt strategies, such as the CoT and PoT, have been proposed to enhance the model's performance in solving descriptive mathematical problems. CoT improves problem-solving accuracy by solving problems step by step [2]. In contrast to CoT, PoT involves generating Python programs during the CoT process and running the resultant program through a Python interpreter to derive the answer [3]. Therefore, PoT has been posited to handle large numbers better than CoT.

Through two distinct experiments, this paper examines whether chatGPT understands place values in numbers using PoT and CoT. In the first experiment, we examine whether chatGPT can correctly order three to six number elements for two types of multipliers, source, and permutation. Here a source multiplier is an array that consists of three to six number elements, which are base 1000 numbers. A permutation multiplier is an array of six elements, which are created from the permutations of the source elements. In the second experiment, we examine whether transforming numerical expressions to English expressions can aid with math word problem solving using chatGPT.

The findings of this paper include the following: **1.** We designed and conducted an experiment and observed that the ordering accuracy for the permutation multipliers was lower than that of source multipliers. **2.** We observed that the GPT based model that utilizes English expressions (CoT_Eng and PoT_Eng), rather than numerical expressions (CoT_Num and PoT_Num) can yield better performance in solving math word problems.

2. Related Works

The study of solving math word problems has been an ongoing field of research since the 1960s [6]. This field has seen progress from rule-based pattern matching algorithms [7, 8] to statistical methodologies [9, 10], tree-based methodologies [11], and has also utilized deep learning-based methodologies [11, 12]. More recently, a significant performance improvement has been observed in studies using pre-trained large language models (LLMs) that adopt the transformer architecture. LLMs are composed of encoder and decoder parts, which are separated from the transformer architecture. Such LLM models have two main approaches: natural language understanding (NLU) models [13] are encoder based approaches, and natural language generation (NLG) models [14] are based on decoders. Between these two approaches, we examine the NLG model approach using the GPT based model.

The advancement in pre-training and the introduction of instructGPT [15] have allowed researchers to provide direct instructions to LLMs. Such models seem to provide reasoning capabilities such as arithmetic [16], commonsense [17, 18], and symbolic reasoning [19]. Through

the implementation of the CoT [2], reasoning becomes possible via ‘natural language rationales’. Furthermore, by leveraging diverse prompt strategies, the gap between human and machine intelligence has been significantly narrowed [20]. The CoT approach encourages the problem-solving approach to be carried out step by step, rather than directly deriving the answer. On the other hand, although the PoT approach follows such a step-by-step approach, the PoT approach generates a Python program. The produced program is then executed via a Python interpreter to compute an answer. Still, the PoT approach has been shown to be susceptible to arithmetic errors. Upon inspection of the problem-solving approach used in each strategy, we noticed that when the text input to GPT is processed, the numbers are divided into multiple tokens. Such division could result in information loss since the original input number could be divided into multiple pieces without preserving the place value information. Thus, in this paper, we designed an experiment to examine whether such information loss occurs by comparing chatGPT inputs of small numbers versus large numbers. We suspect that a performance decrease would occur during math word problem solving when large numbers are tokenized to multiple tokens, and the place value of numbers is not preserved as a result.

3. Method

We devised two experiments to examine whether chatGPT understands the concept of place value, and to see if increasing its understanding of place value through the use of English expressions can help solve math problems.

3.1. Experimental Setup

In this paper, we conducted two experiments to address the following research question: Does chatGPT comprehend the place value in numbers when solving math word problems?

Implementation details. We implemented the following experimental setup, which was based on existing studies that solve math word problems using chatGPT. The GPT3.5 model updates with each model iteration, yielding different results [21]. For the baseline models, we used CoT and PoT, which yield the state-of-the-art performance in SVAMP dataset, using in-context learning with few-shot prompts. For the prompts, we reused those employed in previous studies, utilizing eight examples from the CoT and seven from the PoT [2, 3]. For our newly designed experiment, we modified the number of existing few-shot prompts. For the chatGPT API, we used version gpt3.5-turbo, which is affordable and currently utilized in the chatGPT [21]. Python 3.9 and the Sympy library were used to execute the generated programs. We followed the approach of the prior study [3], setting the temperature to 0 for all experiments excluding self-consistency. For the self-consistency experiments, we set the temperature to 0.7.

3.2. Experiment 1: Ordering Multiplets When the Digits in a Given Number are Reordered

We designed an experiment to assess whether chatGPT accurately perceives the variations in place value in different sequences of numbers. Specifically, we asked chatGPT to order the elements of two sets of multiplets, source, and permutation, in terms of size. A source multiplet

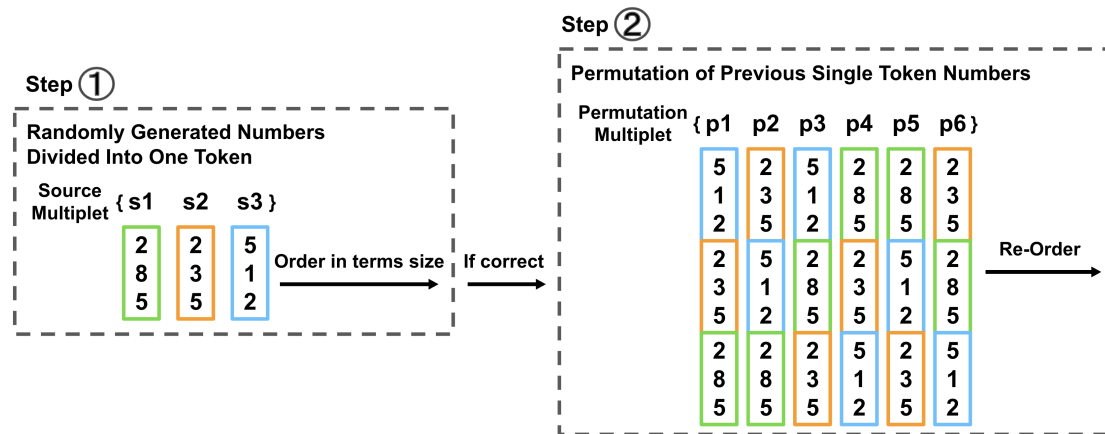


Figure 1: Experiment 1 Setup

is an array of three to six elements, which are 3-digit numbers in base 1000. A permutation multiplet is an array of six elements, which are created from the permutations of the source elements. We tested multiplets of varying lengths, ranging from three to six elements. The experiment is conducted in two steps: (1) ordering elements of source multiplets to examine whether chatGPT can order number that consist of a single token. (2) ordering elements of permutation multiplets to examine whether chatGPT can order numbers that consist of multiple tokens. If chatGPT successfully orders the source multiplet in the first step, a permutation multiplet is created from permutations of the source multiplet. As a permutation multiplet is an array of source multiplets, the elements consisting of a permutation multiplet are divided into multiple tokens. Figure 1 graphically illustrates these two experimental steps when the number of elements consisting of the source multiplet is three.

Note that for experiment 1, we used randomly selected numbers between 100 and 520, rather than 1 and 520, as elements for the source multiplets. Two reasons for such selections are as follows. First, we wanted to use numbers that consist of a single token as an input to chatGPT. Since numbers from 1 to 520 consist of a single token when using OpenAI's tokenizer, the maximum number is set to be 520. Secondly, we set the minimum number to be 100, rather than 1 because the length of the permutation multiplets should be long enough when permutation numbers are generated from the source numbers. Since we want to examine whether chatGPT understands the concept of place value, we used base 1000 numbers, rather than base 10 numbers. Therefore, s1 in Figure 1 has a numerical value of 285, and a place value of 0 when using base 1000. Similarly, p1, the first element of the permutation multiplet in Figure 1, has a numerical value of 512, 235, 285 and corresponding place values are 2, 1, 0.

Ordering the numbers processed as a single token (Step 1 in Figure 1) : In the first step, we examined whether chatGPT accurately identifies the size of numbers that are split into a single token. We randomly generate 3-6 numbers between 100 and 520 (as denoted by source multiplet in Figure 1) to examine whether chatGPT can order them in terms of size. Note that even without understanding the concept of a place value, the relationship for the relative size of single token numbers can be learned. For example, if we have variables A, B, and C, and

data comparing their sizes, we can make size comparisons even without knowing their actual values. Therefore, our goal of this experiment was to check whether chatGPT can compare the sizes of basic unit numbers based on the understanding of place values. The first step is repeated for a total of 1000 trials. Next, we compute “correctly ordered trials” by counting the number of times when the 3-6 numbers are correctly ordered. Finally, to compute the “ordering accuracy” of step 1, we divide the number of correctly ordered trials by 1000.

Ordering the numbers processed as multiple tokens (Step 2 in Figure 1): In step 2, we examined whether chatGPT understands the place value by ordering permuted numbers p_1 - p_6 , which are created using elements s_1 - s_3 from step 1. Only the correctly ordered course multiplets from step 1 are used to create permuted numbers p_1 - p_6 . This is based on the assumption that since chatGPT knows the order of the original elements s_1 - s_3 , the resulting permutations of s_1 - s_3 should also be understood if the model understands the concept of place value. For instance, as shown in Figure 1, p_1 is a permutation of s_1 - s_3 in s_3 - s_2 - s_1 order and p_2 is a permutation listed in s_2 - s_3 - s_1 order. The actual value of p_1 can be understood as the form of $p_1 = s_3 \times 1000^2 + s_2 \times 1000^1 + s_1 \times 1000^0$. Since the maximum number of permutations made using 4-6 elements exceeds six, we randomly select six elements. Next, we use chatGPT to order permutation multiplets in terms of size. If chatGPT understands the concept of place values, it should be able to order the numbers created by permutations of source multiplet in the correct order. We also compute “correctly ordered trials” by counting the number of times when the permutation multiplets are correctly ordered at the second step. Since step 2 is performed only after correctly ordering the numbers in step 1, to calculate the “ordering accuracy” of step 2, we divide the “correctly ordered trials” of step 2 by the “correctly ordered trials” of step 1.

3.3. Experiment 2: Substituting Numerical Expressions with English Expressions

In the second task, we examine whether transforming Numerical expressions to English expressions can aid with math word problem solving using chatGPT. Our hypothesis is that English expression can be used as a method for explicitly providing place value information of a number to chatGPT. To test our hypothesis, we compared the two versions of CoT and PoT: the models with numerical expressions are CoT_Num, PoT_Num and the models with English expressions CoT_Eng, PoT_Eng. We also measure the model’s performance in terms of accuracy when applying the self-consistency, which uses the most frequently asked answer out of five attempts.

For the dataset, we utilized the benchmark dataset of Simple Variations on Arithmetic Math word Problems(SVAMP). The SVAMP dataset is composed from two datasets: 1) the ASDiv-A dataset [22], which requires solving problems in multiple steps, and 2) the MAWPS dataset [23], which consists of arithmetic and algebra problems.

Figure 2 illustrates the prompts and SVAMP dataset examples that are used in experiment 2. The upper two figures represent CoT prompt examples and the lower two figures represent PoT prompt examples. The upper left-side example represents the CoT_Num, which is the original CoT. The upper right-side example illustrates the CoT with English expressions(CoT_Eng), where the Numerical expressions in the CoT_Num prompts and SVAMP dataset have been replaced with English expressions. Note that the numbers in English expressions are capitalized, as in “Seventy Six”. Capitalization was applied to each word representing a numerical unit

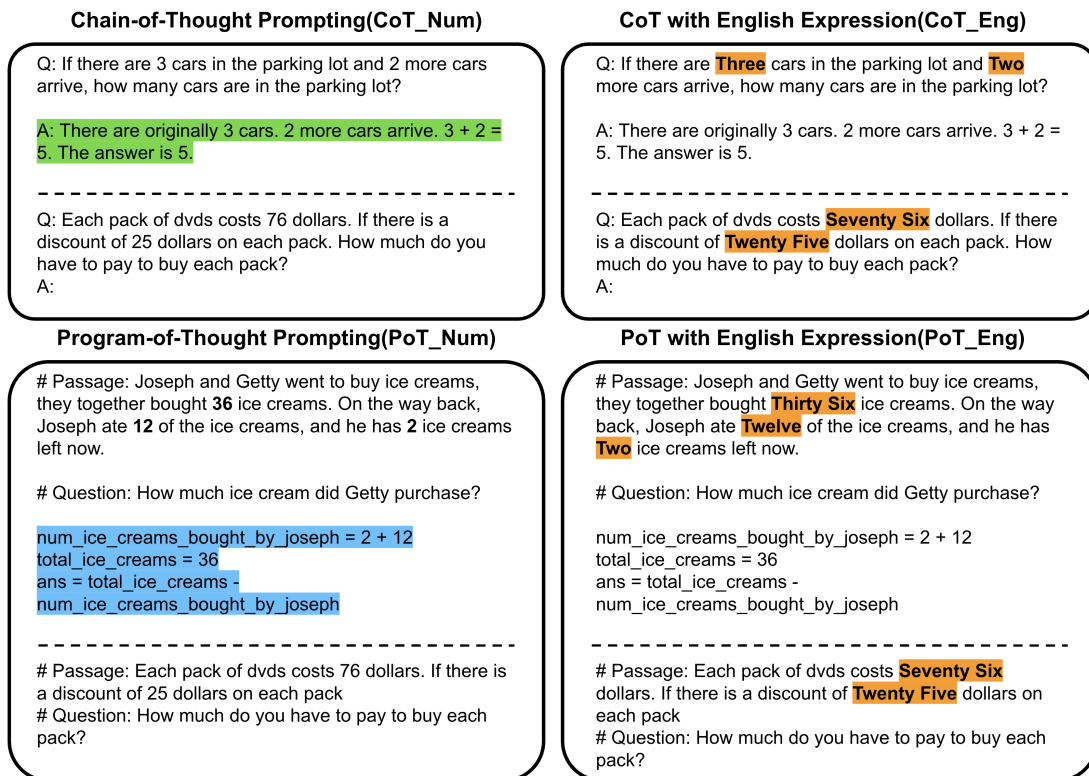


Figure 2: Examples of CoT_Num, CoT_Eng, PoT_Num, PoT_Eng prompts

so that during the tokenization process, each word would be grouped into one token without division.

The lower left-side example represents the Original PoT(PoT_Num) and the lower right-side example represents PoT with English expressions(PoT_Eng). In the PoT_Num example, the number 36, which is composed of the digits 3 and 6, should be interpreted as a sum of 30 and 6, if the model understands the concept of a place value. In the PoT_Eng prompt, such an implicit interpretation is not needed, since the expression “Thirty Six” explicitly delivers the concept of place values. Therefore, we anticipate improved performance in CoT_Eng and PoT_Eng over CoT_Num and PoT_Num respectively.

4. Results and Discussions

4.1. Experiment 1: Ordering Multiplets When the Digits in a Given Number are Reordered

Figure 3 illustrates the outcomes of Experiment 1. Figure 3 (Left) represents two actual queries to the chatGPT using three elements as the source multiplet. In the top section, which shows a sample query for step 1, the chatGPT correctly orders 3 given numbers. Since the multiplet

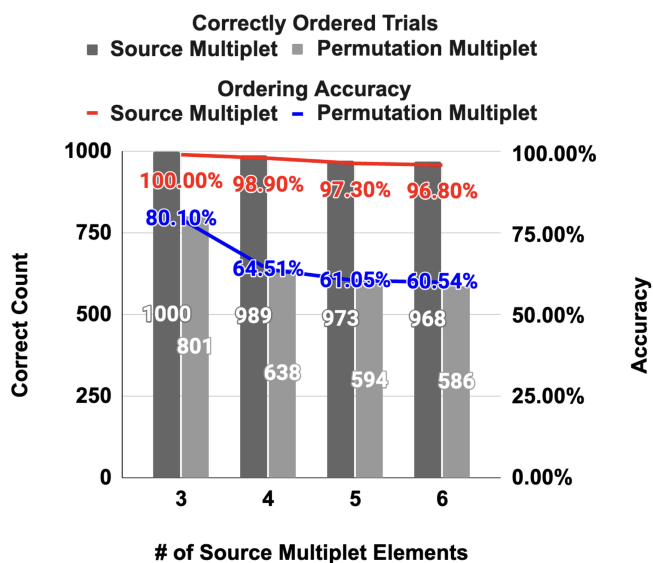
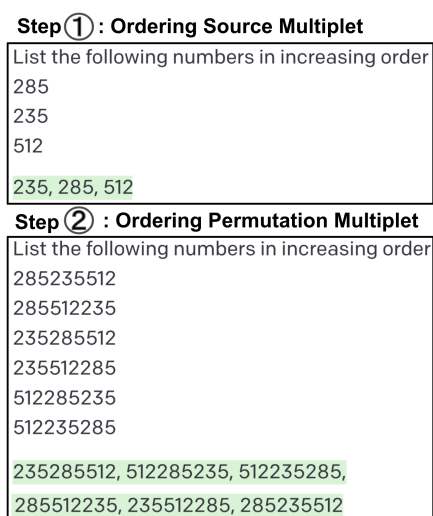


Figure 3: (Left) An example task of ordering source and permutation multiplets. (Right) The “correctly ordered trials” and the “ordering accuracy” for 3-6 elements in multiplets

is ordered in correct order, six numbers are produced from the permutations of the source multipler. In the bottom section, which shows a sample query for step 2, chatGPT is instructed to order the resulting permutation multipler, but failed to order the elements correctly.

Figure 3(Right) shows the results of experiment 1. The x-axis of the graph represents the count of source multipler elements used in the experiment. The bar graph and its y-axis labeled on the left, indicate the number of correctly ordered trials. The line graph and its y-axis labeled on the right, represent the ordering accuracy. Step 2 involves re-ordering the six permutation multiplets that were created using the correctly ordered source multiplets from Step 1. In Figure 3(Right), in the case of ordering source multiplets, the accuracy is decreased slightly from 100.00% to 96.80% as the number of elements increases. On the other hand, when ordering permutation multiplets, accuracy drops significantly from 80.10% to 60.54%. When the ordering accuracy is compared to source multiplets that contain the same number of elements(6 elements), the ordering accuracy for permutation multiplets is lower. Taken together, our experimental results imply that the current version of chatGPT may not fully understand the concept of place value.

4.2. Experiment 2: Substituting Numerical Expressions with English Expressions

Table 1 presents the results of an experiment that involves converting Numerical expressions into English expressions. The self-consistency results in the table were derived from the most frequently produced outcome among five trials at a temperature setting of 0.7. In all cases, the model accuracy when using English expressions(CoT_Eng, PoT_Eng) exceed those of using Numeric expressions(CoT_Num, PoT_Num). There are two possible explanations for the better

Table 1

CoT, PoT_Num, PoT_Eng results using original SVAMP dataset with gpt3.5-turbo

	Temperature	CoT_Num	CoT_Eng	PoT_Num	PoT_Eng
Accuracy	0.0	76.80	79.90	80.04	82.00
Accuracy (SC)	0.7	81.20	82.50	82.60	83.70

performance of models that use English expressions. Firstly, since the English expressions explicitly incorporate the concept of place value, such expressions hold an advantage over Numerical expressions, which implicitly contain the concept. As a result, substituting Numerical expressions with English expressions could help with solving mathematical problems. Secondly, a performance increase may have been simply because chatGPT is trained to handle natural language better than numbers. To address this second possibility, a model that can deliver place value information without using English expressions should be tested.

Compared to the performance increase of PoT Eng over PoT Num(1.96%), the performance increase of CoT Eng over CoT Num(3.10%) is greater. A possible explanation for a higher performance of the PoT over CoT is the use of Python interpreter by PoT. In order to solve math problems correctly, CoT needs to understand the word problem, create a solution and compute the correct answer directly. In the case of PoT, understanding the problem and creating a solution process is also required as in the case of CoT. However, the numbers used in the problem are put into variables, and the calculation is done by the Python interpreter. Therefore, compared to PoT, the CoT model may be affected more by the missing information, which involves understanding the place value.

5. Conclusion

Our experimental result presents pieces of evidence that provide support for our initial question on whether chatGPT comprehends the place value system in numbers when solving math word problems. The results of the first experiment showed that the ordering accuracy for the permutation multipliers was lower than the ordering accuracy for the source multipliers. Furthermore, as the number of elements in the source multiplier increases, the difference in the ordering accuracy between the source and permutation multiplier increases. Our second experiment showed that the accuracy in solving arithmetic problems could be enhanced when it understands the place value of numbers.

The Limitations of the two experiments are as follows. For the first experiment, our experiment only shows results with three-digit tokens in base 1000 to show that the chatGPT may not comprehend the concept of place value. However, since single or double-digit numbers frequently appear in math word problems, experiments with smaller digits should be conducted to examine whether a similar phenomenon occurs for tokens with different lengths. For the second experiment, the numbers in the SVAMP dataset are relatively small, either single or double digits. Therefore, the experimental results may not replicate when numbers get larger. Thus, future experiments with numbers larger than three digits should be conducted to fully examine the phenomenon.

Despite such limitations, the observations from both experiments suggest that the concept of place value may not be adequately integrated when numbers are represented as tokens in chatGPT. Therefore, if the numbers in the chatGPT model can be trained to understand the concept of place value despite being split into multiple tokens, the model may yield a higher performance in solving math word problems or problems that involve an understanding of place values in numbers. For example, engineering prompts to provide explicit definitions of place values or fine-tuning a language model extensively with numbers may be possible directions to pursue.

6. Acknowledgement

This work was supported by the National Research Foundation of Korea (NRF) grant (No. 2020R1C1C1010162) funded by the Korean government (MSIT).

References

- [1] S. T. Pham, P. M. Sampson, The development of artificial intelligence in education: A review in context, *Journal of Computer Assisted Learning* 38 (2022) 1408–1421.
- [2] J. Wei, X. Wang, D. Schuurmans, M. Bosma, E. Chi, Q. Le, D. Zhou, Chain of thought prompting elicits reasoning in large language models, *arXiv preprint arXiv:2201.11903* (2022).
- [3] W. Chen, X. Ma, X. Wang, W. W. Cohen, Program of thoughts prompting: Disentangling computation from reasoning for numerical reasoning tasks, *arXiv preprint arXiv:2211.12588* (2022).
- [4] L. Verschaffel, S. Schukajlow, J. Star, W. Van Dooren, Word problems in mathematics education: A survey, *ZDM* 52 (2020) 1–16.
- [5] M. Firat, How chat gpt can transform autodidactic experiences and open education, Department of Distance Education, Open Education Faculty, Anadolu Unive (2023).
- [6] D. Bobrow, et al., Natural language input for a computer problem solving system (1964).
- [7] C. R. Fletcher, Understanding and solving arithmetic word problems: A computer simulation, *Behavior Research Methods, Instruments, & Computers* 17 (1985) 565–571.
- [8] M. Yuhui, Z. Ying, C. Guangzuo, R. Yun, H. 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, IEEE, 2010, pp. 476–479.
- [9] M. J. Hosseini, H. Hajishirzi, O. Etzioni, N. Kushman, Learning to solve arithmetic word problems with verb categorization., in: *EMNLP*, 2014, pp. 523–533.
- [10] A. Mitra, C. Baral, Learning to use formulas to solve simple arithmetic problems, in: *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2016, pp. 2144–2153.
- [11] L. Wang, D. Zhang, L. Gao, J. Song, L. Guo, H. T. Shen, Mathdqn: Solving arithmetic word problems via deep reinforcement learning, in: *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

- [12] B. Kim, K. S. Ki, D. Lee, G. Gweon, 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), 2020, pp. 3768–3779.
- [13] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, Bert: Pre-training of deep bidirectional transformers for language understanding, arXiv preprint arXiv:1810.04805 (2018).
- [14] A. Radford, K. Narasimhan, T. Salimans, I. Sutskever, et al., Improving language understanding by generative pre-training (2018).
- [15] L. Ouyang, J. Wu, X. Jiang, D. Almeida, C. Wainwright, P. Mishkin, C. Zhang, S. Agarwal, K. Slama, A. Ray, et al., Training language models to follow instructions with human feedback, Advances in Neural Information Processing Systems 35 (2022) 27730–27744.
- [16] A. Lewkowycz, A. Andreassen, D. Dohan, E. Dyer, H. Michalewski, V. Ramasesh, A. Slone, C. Anil, I. Schlag, T. Gutman-Solo, et al., Solving quantitative reasoning problems with language models, arXiv preprint arXiv:2206.14858 (2022).
- [17] J. Jung, L. Qin, S. Welleck, F. Brahman, C. Bhagavatula, R. L. Bras, Y. Choi, Maieutic prompting: Logically consistent reasoning with recursive explanations, arXiv preprint arXiv:2205.11822 (2022).
- [18] J. Liu, S. Hallinan, X. Lu, P. He, S. Welleck, H. Hajishirzi, Y. Choi, Rainier: Reinforced knowledge introspector for commonsense question answering, arXiv preprint arXiv:2210.03078 (2022).
- [19] D. Zhou, N. Schärli, L. Hou, J. Wei, N. Scales, X. Wang, D. Schuurmans, O. Bousquet, Q. Le, E. Chi, Least-to-most prompting enables complex reasoning in large language models, arXiv preprint arXiv:2205.10625 (2022).
- [20] S. Qiao, Y. Ou, N. Zhang, X. Chen, Y. Yao, S. Deng, C. Tan, F. Huang, H. Chen, Reasoning with language model prompting: A survey, arXiv preprint arXiv:2212.09597 (2022).
- [21] OpenAI, Chatgpt, Available at <https://openai.com/blog/chatgpt/> (2023/05/15), ????
- [22] S.-Y. Miao, C.-C. Liang, K.-Y. Su, A diverse corpus for evaluating and developing english math word problem solvers, arXiv preprint arXiv:2106.15772 (2021).
- [23] R. Koncel-Kedziorski, S. Roy, A. Amini, N. Kushman, H. Hajishirzi, 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, 2016, pp. 1152–1157.