# Applying Mixture of Experts Technique on Small-Size Language Models for Multi-Step Mathematical Reasoning Problems

Daiwei Zhang* Tianyang Xu* Yaqi Qin* Zhiyi Chen*

## Abstract

While large language models (LLM) have demonstrated state-of-the-art performance in many NLP tasks, it is often time and resource consuming to train a large-scale model with billions of tokens, creating barrier for academic researchers and small companies to train their own model. It is verified experimentally that scaling up language models can introduce emergent abilities, which do not exist in smaller models. On the other hand, specialized prompting methods such as Chain-of-Thought are not effective unless the model size exceeds a specific level (Wei et al., 2022a). Thus, we apply the Reasoning by Asking method (Shridhar et al., 2023), focus on the specific task of multi-step mathematical reasoning, and improve the performance of smaller language models using a Mixture-of-Experts (MoE) technique. We demonstrate that by fine-tuning a GPT2 or DialoGPT model for each smaller problem space, our method achieve a higher accuracy on the GSM8K dataset than fine-tuning a single model. We also compared and discussed the performance between these base models in this report.

## 1 Introduction

Despite the vast improvements brought by pre-trained Large Language Models in many natural language processing tasks including text translation and language modeling, its large number of parameters and pre-train data do not lead to high performances on challenging tasks, such as arithmetic, commonsense, and symbolic reasoning (Rae et al., 2021). Step-by-step reasoning approaches like Chain-of-Thought (CoT) have proven to be efficient when inducing the reasoning capabilities in large language models (Wei et al., 2022b), however the performance of these approaches largely depends on the model size.

Our goal is to use Mixture of Exports (MoE) techniques on smaller-size language models to compensate for the resources needed to train or call the API of a larger language model in a multi-hop reasoning process.

In this project we focus on solving Math Word Problems (MWP). To apply MoE, we first need to divide the problem space into disjoint sub-spaces, and train a model for each sub-space. For multi-step problems, we train a model for problems with specific number of steps to solve, which ranges from 2 to 8 in GSM8K dataset. Number of steps needed to solve a problem often correspond to the complexity of a problem. In the two-stage experiments we conducted, we have fine-tuned a GPT2-small or DialoGPT-small (Zhang et al., 2019) model for each subset of questions on the step-by-step solution procedure provided by the GSM8K dataset. We then compared their performance at validation and inference time with the baseline model, which is trained on all training data. [1]

## 2 Related work

**Math Word Problems (MWP)** The task of Mathematical Word Problem Solvers (MWPs) is to automatically answer math questions based on a concise problem description. An example is provided in Table 1. Numerous researchers have conducted investigations in this field, leading to the introduction of various approaches, including rule-based (Kintsch and Greeno, 1985) methods, statistical machine learning (Kushman et al., 2014; Roy and Roth, 2018) techniques, and Seq2Seq-based approaches (Wang et al., 2018). Wang et al. attempted to try decomposing the operations needed to solve the problem into an expression tree. Xie and Sun later explored a goal-driven approach to decompose the problem into the sub-questions and construct an expression tree using these sub-questions.

With the advancements in large language models (LLMs), several approaches have been proposed and have demonstrated significant improvements. One notable approach is the Chain-of-Thought (CoT) approach (Wei et al., 2022b), which enhances the reasoning ability of LLMs by prompting

[1] Code and preprocessed data are available at https://github.com/sally-xu-42/CSNLP_MoE_MathReasoning

D. Zhang, T. Xu, Y. Qin, Z. Chen are with ETH Zurich {daizhang,tianyxu,yaqqin,zhiychen}@ethz.ch.
* indicates equal contribution.

them with a series of intermediate reasoning steps. However, applying CoT prompting to smaller scale models is not feasible (Wei et al., 2022b).

An alternative approach that shows potential is knowledge distillation. It involves using the CoT outputs of LLMs as training data for smaller models, aiming to transfer the reasoning ability to these models. Unfortunately, this attempt has proven unsuccessful due to the limited reasoning capability of small models (Stolfo et al., 2022). Recently, the Decompositional Distillation approach has been proposed as an alternative (Shridhar et al., 2022). Instead of training small models with CoT results from LLMs, this approach decomposes the input problem into a series of subquestion-solution pairs. The small models trained using this approach have shown superior performance compared to models of the same size trained to reproduce CoT results, achieving a 35% improvement on the GSM8K dataset.

**Mixture of Experts** Mixture of Experts (MoE) was first introduced long ago (Jacobs et al., 1991). The essential idea is to construct a collection of experts and each learns from a subset of the training samples. Various types of expert architecures have been proposed, such as SVM, Gaussian processes and deep networks.

This method was introduced to solve learning problems regarding natural languages (e.g. language modeling and machine translation tasks) (Shazeer et al., 2017). Due to the increase in the size of dataset, larger and larger model capacity is needed, which leads to explosion in training cost. The goal of applying MoE is to decrease the total number of updates on all parameters, since the parameters of each expert are only updated by the corresponding subset of training samples.

**Response Generation** Recently many improvements in different neural language tasks have been made using large-scale models based on transformers (Radford et al., 2018; Raffel et al., 2019). Thanks to the self-attention mechanism in the transformer-based models, they are capable of capturing long-term dependencies in textual data (Vaswani et al., 2017). These models, such as GPT-2 (Radford et al., 2018) or BERT (Devlin et al., 2018) are usually pre-trained on large-scale data to learn to capture the relations between tokens and sentences. GPT-2 has shown good performance in generating fluent and diverse responses. However, since most such models are pre-trained on general text data, they sometimes generate undesirable repetition and monotone answers. Other models, such as DialoGPT (Zhang et al., 2019) and DLGnet (Olabiyi and Mueller, 2019) are later proposed for generating more relevant and context-consistent responses in dialog settings. The data used in their pre-training processes are dialogs. For instance in DialoGPT, bthe data used for pre-training is chains of comments collected from Reddit. To avoid repetition and bland responses, filtering of the pre-train data by removing repetitive and uninformative target responses is often included and mutual informative maximization are used when pre-training DialoGPT. The later achieves better results on DSTC-7 (Dialogue Generation Challenge) when compared with GPT-2.

## 3 Dataset

The reasoning dataset we adopted is Socratic version of GSM8K dataset (Cobbe et al., 2021). It consists of 8.5K multi-step grade school math word problems written by human writers, while the number of reasoning steps vary from 2 to 8. Getting the ultimate solutions to the given problems involve executing a series of elementary arithmetic operations ($+ - \times \div$). These problems are designed to be within the knowledge range of a proficient middle school student, thus they serve as an effective measure for developing multi-step mathematical reasoning abilities in language models.
The dataset is originally segmented into 7.5K training problems and 1K testing problems, which we followed in our experiment settings. The approach we used to split the data for each sub-model is: we first calculated the number of reasoning steps for each MWP record, then for each specific number of steps, we took the MWP records with num_steps larger or equal than that and pruned them to the number of steps we require.

After the preprocessing described above, size of training set, validation set and test set for each sub-model are listed in Table 2.

## 4 Experiments

We conducted two studies using different models to compare the performances of MoE and single model. Study 1 used GPT2-small model for each number-of-step in the GSM8K dataset and Study 2 used DialoGPT as the base model. Due to the divergent nature of the two models. We used dif-

**Q:** A robe takes 2 bolts of blue fiber and half that much white fiber. How many bolts in total does it take?
**A:** The answer is 3

Table 1: Example of MWPs

| Number of Steps | Training Size | Validation Size | Testing Size |
|---|---|---|---|
| 2 steps | 7473 | 748 | 326 |
| 3 steps | 5513 | 552 | 370 |
| 4 steps | 3027 | 337 | 298 |
| 5 steps | 1533 | 171 | 174 |
| 6 steps | 750 | 84 | 88 |
| 7 steps | 331 | – | 40 |
| 8 steps | 103 | – | 20 |

Table 2: Size of training, validation and testing subsets with specific number of steps. Currently we did not train model with 7 and 8 steps because the training subset is too small, and our model has too many parameters, causing them to overfit easily.

ferent data preprocessing methods as introduced below.

Note that we followed one of the settings introduced in Shridhar et al., that is, all the sub-questions are expected to be generated by a LLM-based question-generation (QG) model but can also be given as ground-truth as the best-case senario to more effectively construct and evaluate a separate question-answering (QA) model we want to focus on. Hence in all our experiments, the sub-questions are directly adapted from the Socratic version of GSM8K dataset and given as input in both training and evaluation.

### 4.1 Study 1 experiment setting

- **Data preprocessing.** To construct the input data for fine-tuning language models, for each math problem, we extracted the context $C$, the main problem $P$ and each sub-question and answer pairs $(q_i, a_i)$. Since we wanted to use questions as the prompt to get all the corresponding answers, we first concatenated all the sub questions together with $C$ and $P$, and put all the sub answers at the end of the sequence.
  Following the tradition of language modelling, we added one $\langle BOS \rangle$ token to mark the beginning of the sequence, and one $\langle EOS \rangle$ token at the end to indicate termination of the generation process. In addition, to separate question segments from the answer segments, we inserted one $\langle SEP \rangle$ token after the last sub question. The intuition is to indicate the model to start generating answers. One example of the final constructed input is shown in Figure 1.a.

- **Train.** For each problem complexity (number of reasoning steps ranges from 2 to 6), we trained one expert model. Here we excluded 7 and 8 steps due to lack of data (each with around 100 records). We also trained one baseline model that uses all the training data with various steps. All models were fine tuned based on the small sized GPT2 using the Huggingface library. To avoid overfitting, all the models were trained for 50 epochs with early stop. We also set batch size = 30 and lr = 5e-4.

- **Inference.** During inference time, we first dispatched the inference job to the corresponding expert model based on the number of sub questions. The prompt text we used is constructed as $\langle BOS \rangle + C + P + q_1 + ... + q_n + \langle SEP \rangle$. The expert model was supposed to generate all corresponding sub answers until the $\langle EOS \rangle$ token. Since small models are lacking in the ability of arithmetic calculation, whenever an equation is generated, we manually extracted the expression and used python to get the solution which is then concatenated with the previously generated sequence.

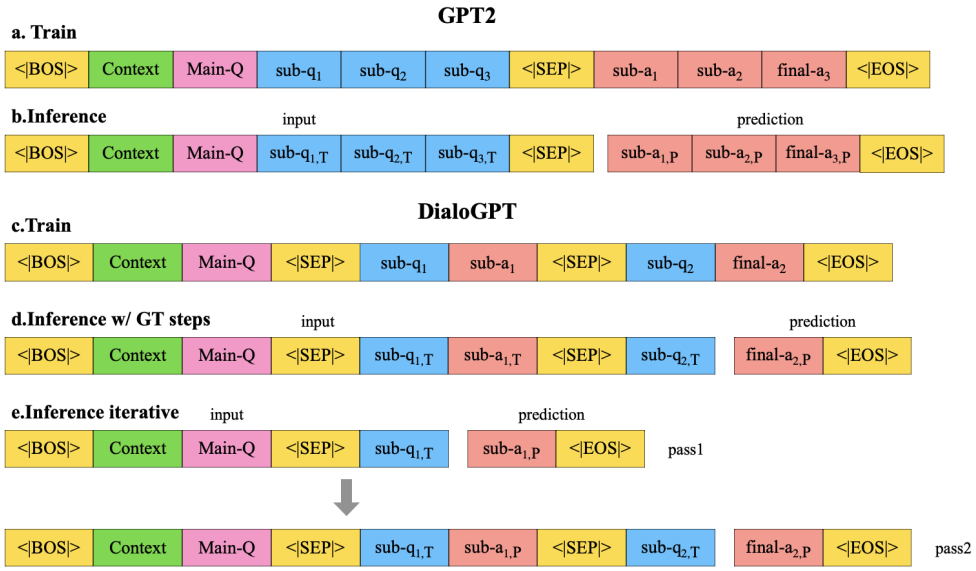


Figure 1: Illustration of how a math problem is composed into input and output in study 1 & 2.

- **Evaluation metrics.** Since the last sub-answer should contain the final solution of the math problem, we calculated the accuracy of the numeric solution generated at the last step to measure the model's ability of multi-step math reasoning. The accuracy of Mixture of Expert (MoE) is simply the overall accuracy of all expert models.

### 4.2 Study 2 experiment setting

- **Data preprocessing.** In study 2, we focused on switching the base model from GPT-2 to DialoGPT. In order to do this, we needed to change the input data to a dialogue format. For each math problem, we concatenated the data entry in the following order during training: context $C$, the main problem $P$, and each sub-question and answer pairs $(q_i, a_i)$. The special tokens are the same from study 1. One example of the final constructed input is given in Figure 1.c.

- **Train.** The training procedure was the same as study 1, we trained one expert model for each problem complexity, except for all models were fine tuned based on DialoGPT model (Zhang et al., 2020). The other hyperparameters remained the same.

- **Inference.** Given the nature of dialog-like data, we cannot infer in the same setting as study 1; on the other hand, we proposed two methods for evaluating the performance of the model in two distinct settings.

  The first is illustrated in Figure 1.d, It simply predicts the last sub-answer, which is also the answer of main problem, given all the previous reasoning steps and corresponding sub-question ($\langle BOS\rangle + C + P + q_1 + a_1 ... + q_n + \langle SEP\rangle$). We expected such expert model to perform better than the model from study 1 when handling the same set of math problems since more information (all the previous sub-answers) is given to the model.

  The second method is an iterative approach, illustrated in Figure 1.e. In the first pass, context, main question, and the first sub-question is concatenated as the input ($\langle BOS\rangle + C + P + q_1$) for the expert model which specifically handling problems with $N_{\text{steps}} = 1$; the LLM-generated sub-answer $a_1$, with $\langle EOS\rangle$ replaced by $\langle SEP\rangle$, and the next ground-truth sub-question $q_2$ are then appended to the previous input. Such new input $\langle BOS\rangle + C + P + q_1 + a_1 + \langle SEP\rangle + q_2$ is fed into the next expert model handling $N_{\text{steps}} = 2$, and so on until the last sub-answer $a_N$ is generated corresponding to $q_N$. This sub-answer is the final answer for the whole math problem.

- **Evaluation metrics.** The evaluation metric is same as study 1, while we only considered the accuracy of the numerical solution at the last step, which is also the main answer, and

**Context:** John visits his parents twice a month. It takes him 2 hours to drive there at a speed of 70 mph.
**Main-Q:** Considering the round trip, how many miles a month does he drive when visiting his parents?
**Sub-QA1:** How far away are his parents? His parents live 70∗2=<<70∗2=140>>140 miles away.
**Sub-QA2:** How many miles does he drive in total? So he drives 140∗2=<<140∗2=280>>280 miles a round trip.
**Sub-QA3:** How many miles does he drive in total? 280∗2 = <<280∗2=560>>560 miles a month.
**Answer:** 560
**LLM-generated:** <|BOS|>John visits .... It takes .... Considering the round trip, how many miles a month does he drive when visiting his parents?<|SEP|> How far away are his parents? He drives 2∗70=<<2∗70=140>>140 miles away.<|SEP|> How many miles does he drive in total? So he drives 140+140=<<140+140=280>>280 miles in total.<|SEP|> How many miles does he drive in total? He drives 280+280=<<280+280=560>>560 miles in total.<|EOS|>

Table 3: Illustration of one preprocessed GSM8K problem with $N_{\text{steps}} = 3$ and its corresponding DialoGPT-generated answers by the iterative inference method described in Figure 1.e. The orange text is the ground-truth sub-answers; the blue text represents the LLM-generated answers in three iterations. Some parts of the duplicated context are omitted for simplicity.

| | GPT2 | DialoGPT | |
|---|---|---|---|
| $N_{\text{steps}}$ | $\text{Acc}_{\text{final-a w/ GT sub-q}}$ | $\text{Acc}_{\text{sub-a w/ GT steps}}$ | $\text{Acc}_{\text{final-a iterative}}$ |
| 2 | 11.90 | 7.67 | 5.52 |
| 3 | 5.68 | 7.57 | 1.89 |
| 4 | 2.01 | 11.07 | 1.68 |
| 5 | 0.58 | 2.30 | 0.00 |
| 6 | 0.00 | 0.00 | 0.00 |
| MoE | 5.33 | 6.82 | 2.27 |
| Baseline w/o MoE | 5.05 (Shridhar et al., 2023) | 1.67 | - |

Table 4: Accuracy (in %) of GPT2-based and DialoGPT-based expert models, MoE, or baseline models evaluated on GSM8K test set. $\text{Acc}_{\text{final-a w/ GT sub-q}}$ represents the accuracy of the final answer by the similar inference process as Figure 1.b; $\text{Acc}_{\text{sub-a w/ GT steps}}$ is the accuracy of the last sub-answer, which is also the final answer in most cases, given all the previous ground-truth steps (sub-q&a pairs) and its corresponding sub-question (Figure 1.d); $\text{Acc}_{\text{final-a iterative}}$ is the final accuracy obtained by iterative inference (Figure 1.e). Accuracy of MoE is the weighted average of all expert models on the test set. Each number $N_{\text{steps}}$ indicates inferencing on problems with this number of reasoning steps, corresponding to a specific expert model in the first two types of evaluation.

evaluated the accuracy of our MoE method as the $p$-weighted average of all expert models, where $p$ represents the proportion of problems with the $N_{\text{steps}}$ corresponding to that expert in the test set.

### 4.3 Results

Accuracy of each expert models, the Mixture of Experts and the baseline model are listed in Table 4. Note that the accuracy obtained by GPT2 and DialoGPT are not comparable, as distinct ground-truth information was exposed to the trained expert models during the inference as illustrated in Figure 1 (all sub-questions vs. previous reasoning steps). One key observation of this table is that the accuracy generally dropped for expert models handling more complicated problems, that is, requiring more reasoning steps; however, we can not conclude such behavior is due to the complexity of problem itself or the limited training data given the unbalanced distribution of GSM8K illustrated in Table 2.

We did observe improved accuracy by Mixture of Experts compared to a single model. Our GPT2-based MoE outperforms the accuracy reported in (Shridhar et al., 2023), where the GPT2-small model is also trained on ground-truth step-by-step annotations and inference as we did. Similarly, the DialoGPT-based MoE significant outperforms the single model we trained on all math problems from

the training set and inference in a single go.

Finally, we experimented in a more realistic setting with the iterative evaluation. The accuracy drops more dramatically when inferencing on problems with more reasoning steps, as more expert models are involved while iterating through reasoning steps; for example, evaluating a problem with $N_{\text{steps}} = 3$ requires samples from three distinct expert models step by step, hence the incorrect reasoning from previous LLMs can significantly damaged the later generated text. One correct example obtained by such evaluation is illustrated in Table 3.[2]

## 5 Future work

### 5.1 Limitation

Due to the limited size of GSM8K dataset and relatively high complexity of GPT-2 model (8.5k MWPs vs. 124M parameters), fine-tuning such models suffers from overfitting even in its baseline setting, whereas our Mixture-of-Experts approach suffers to a greater degree from this. When training the expert model targeting the problems with 6 steps (750 records), we will have overfitting in few epochs. Besides, the performance of fine-tuned models can differ randomly on the test set.

Another limitation is that using accuracy as the only evaluation metric for inference-time evaluation is a limited representation of model's reasoning skills. Due to the nature of math problems, a calculator is used for inference-time prediction to transform the generated string within the $<<>>$ bracket into a number for exact comparison. Though accuracy is the most straightforward and interpretable evaluation metric for MWPs, we are looking for other metrics or data sources that can more effectively represent model's reasoning capabilities.

### 5.2 Possible Improvements

To address some of the above-mentioned issues, the most preliminary approach is to adapt more MWP datasets, which may not have multi-step mathematical reasoning but has more variations over structure, to increase model's ability to generalize, for example the SVAMP dataset (Patel et al., 2021). Moreover, we are able to augment our dataset through three types of variations for creating SVAMP, e.g. changing the principal object, inverting operations, changing order of objects, or adding irrelevant information. With the augmented dataset, we expect our models to obtain stronger mathematical reasoning abilities.

While we are still searching for other datasets, preferably the mathematical problems which can be manually decomposed into step-by-step procedures and transformed into the so-called Socratic CoT, i.e. a sequence of subproblem-solution pairs, we are attempting to incorporate other non-mathematical datasets with only original problem and learn a semantic decomposition of the original problem, for example the StrategyQA dataset, which consists a factual question with binary True/False as the final answer. To do so we aim to reconstruct the whole framework, including the problem decomposer, proposed in (Shridhar et al., 2022), and therefore we can see how adapting Mixture of Experts (MoE) techniques could improve the whole knowledge distillation pipeline.

What's more, an acute drop in prediction accuracy of more complex problems is observed for both GPT2 and DialoGPT due to extremely unbalanced data distribution. To tackle this problem, we could try to transfer the capability of experts targeting at less complex problems to experts for more complex problems in an iterative way. For instance, we could train the 3-step-expert using the 2-step-expert as the pretrained backbone, and finetune it with all 3-step data. And similarly, we train all the experts iteratively leveraging the less step expert model as the starting points. In this way, the experts targeting at solving longer step reasoning could utilize all data with fewer steps.

## Acknowledgments

We are grateful for the help from our teaching assistant Alessandro Stolfo and Prof. Mrinmaya Sachan throughout this semester.

## References

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. 2021. Training verifiers to solve math word problems.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: pre-training of

[2]The interactive inference pipeline is available at https://colab.research.google.com/drive/1rOKXyNm_6mMfeMOV6nPeugQ0Bwj8Tsjk?usp=sharing

deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805.

Robert A. Jacobs, Michael I. Jordan, Steven J. Nowlan, and Geoffrey E. Hinton. 1991. Adaptive mixtures of local experts. *Neural Computation*, 3(1):79–87.

Walter Kintsch and James G. Greeno. 1985. Understanding and solving word arithmetic problems. *Psychological review*, 92 1:109–29.

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.

Oluwatobi Olabiyi and Erik T. Mueller. 2019. Multi-turn dialogue response generation with autoregressive transformer models. *CoRR*, abs/1908.01841.

Arkil Patel, Satwik Bhattamishra, and Navin Goyal. 2021. Are nlp models really able to solve simple math word problems?

A. Radford, Jeffrey Wu, R. Child, David Luan, Dario Amodei, and Ilya Sutskever. 2018. Language models are unsupervised multitask learners. In *Technical report, OpenAI*.

Jack W. Rae, Sebastian Borgeaud, Trevor Cai, Katie Millican, Jordan Hoffmann, H. Francis Song, John Aslanides, Sarah Henderson, Roman Ring, Susannah Young, Eliza Rutherford, Tom Hennigan, Jacob Menick, Albin Cassirer, Richard Powell, George van den Driessche, Lisa Anne Hendricks, Maribeth Rauh, Po-Sen Huang, Amelia Glaese, Johannes Welbl, Sumanth Dathathri, Saffron Huang, Jonathan Uesato, John Mellor, Irina Higgins, Antonia Creswell, Nat McAleese, Amy Wu, Erich Elsen, Siddhant M. Jayakumar, Elena Buchatskaya, David Budden, Esme Sutherland, Karen Simonyan, Michela Paganini, Laurent Sifre, Lena Martens, Xiang Lorraine Li, Adhiguna Kuncoro, Aida Nematzadeh, Elena Gribovskaya, Domenic Donato, Angeliki Lazaridou, Arthur Mensch, Jean-Baptiste Lespiau, Maria Tsimpoukelli, Nikolai Grigorev, Doug Fritz, Thibault Sottiaux, Mantas Pajarskas, Toby Pohlen, Zhitao Gong, Daniel Toyama, Cyprien de Masson d'Autume, Yujia Li, Tayfun Terzi, Vladimir Mikulik, Igor Babuschkin, Aidan Clark, Diego de Las Casas, Aurelia Guy, Chris Jones, James Bradbury, Matthew J. Johnson, Blake A. Hechtman, Laura Weidinger, Iason Gabriel, William Isaac, Edward Lockhart, Simon Osindero, Laura Rimell, Chris Dyer, Oriol Vinyals, Kareem Ayoub, Jeff Stanway, Lorrayne Bennett, Demis Hassabis, Koray Kavukcuoglu, and Geoffrey Irving. 2021. Scaling language models: Methods, analysis & insights from training gopher. *CoRR*, abs/2112.11446.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2019. Exploring the limits of transfer learning with a unified text-to-text transformer. *CoRR*, abs/1910.10683.

Subhro Roy and Dan Roth. 2018. Mapping to declarative knowledge for word problem solving. *Transactions of the Association for Computational Linguistics*, 6:159–172.

Noam Shazeer, *Azalia Mirhoseini, *Krzysztof Maziarz, Andy Davis, Quoc Le, Geoffrey Hinton, and Jeff Dean. 2017. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. In *International Conference on Learning Representations*.

Kumar Shridhar, Alessandro Stolfo, and Mrinmaya Sachan. 2022. Distilling multi-step reasoning capabilities of large language models into smaller models via semantic decompositions.

Kumar Shridhar, Alessandro Stolfo, and Mrinmaya Sachan. 2023. Distilling reasoning capabilities into smaller language models.

Alessandro Stolfo, Zhijing Jin, Kumar Shridhar, Bernhard Schölkopf, and Mrinmaya Sachan. 2022. A causal framework to quantify the robustness of mathematical reasoning with language models.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *CoRR*, abs/1706.03762.

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.

Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, Ed H. Chi, Tatsunori Hashimoto, Oriol Vinyals, Percy Liang, Jeff Dean, and William Fedus. 2022a. Emergent abilities of large language models.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, brian ichter, Fei Xia, Ed H. Chi, Quoc V Le, and Denny Zhou. 2022b. Chain of thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems*.

Zhipeng Xie and Shichao Sun. 2019. A goal-driven tree-structured neural model for math word problems. In *International Joint Conference on Artificial Intelligence*.

Yizhe Zhang, Siqi Sun, Michel Galley, Yen-Chun Chen, Chris Brockett, Xiang Gao, Jianfeng Gao, Jingjing Liu, and Bill Dolan. 2019. Dialogpt: Large-scale generative pre-training for conversational response generation. *CoRR*, abs/1911.00536.

Yizhe Zhang, Siqi Sun, Michel Galley, Yen-Chun Chen, Chris Brockett, Xiang Gao, Jianfeng Gao, Jingjing Liu, and Bill Dolan. 2020. Dialogpt: Large-scale generative pre-training for conversational response generation.