

教育を目的とした日本語初等数学問題に特化した大規模言語モデルの構築

苗中濤* 趙開顔* 呉梓隆 呉奇宇 鶴岡慶雅

東京大学大学院 情報理工学系研究科

{mzt,zhaokaiyan1006,zw2599,qiyuw,yoshimasa-tsuruoka}@g.ecc.u-tokyo.ac.jp

概要

近年、様々なオープンソース大規模言語モデルの登場に伴い、特定の領域向けの大規模言語モデルを開発することが可能になっている。しかし、日本語での初等数学教育を目的とした大規模言語モデルは未だに存在していない。本稿では、基礎的な数学問題を解き、教育に有用な推論過程が付いた答えが出力できる日本語大規模言語モデルを構築する方法について報告する。具体的には、日本語の初等数学問題データセットを構築し、Japanese Stable LM Beta 7B をファインチューニングし、日本語数学専門言語モデル JEElementaryMathGPT を構築した。さらに、モデルの評価方法について議論し、既存の言語モデルと性能を比較した。

1 はじめに

ChatGPT [1, 2] をはじめとする大規模言語モデルは、高い言語理解能力と言語生成能力を持つことが示されている。大規模言語モデルは翻訳 [3] やコード生成 [4]、QA (質問応答) [5]、文章要約 [6] などの様々なタスクにおいて、その有効性が報告されている [7, 8]。

大規模言語モデルの優れた言語理解力と生成能力を利用して、数学の問題を解くことが試みられているが、現在の大規模言語モデルでは解けない問題が多いことが報告されている。図 1 に示すように、これらの大規模言語モデルは、非常に簡単な数学問題でさえも、解けないことがある [9]。そのため、大規模言語モデルの数学能力を改善する様々な手法が試みられている。例えば、WizardMath [10] は、RLEIF (Reinforcement Learning from Evol-Instruct Feedback) を提案し、Llama 2 [11] の数学推論能力を改善した。Llemma [12] は、科学論文、数学問題や

コードなどを含むウェブデータである Proof-Pile-2 データセットで Code Llama [4] の事前学習を継続し、数学推論能力を向上させた。

一方、大規模言語モデルは教育分野でも徐々にその有効性が示されつつある。例えば、大規模言語モデルは Python や C++、Rust などの初心者のプログラミング言語の学習を助けることができる [13]。また、非英語母国語話者の英語会話やライティングの練習を助けることができる [14]。今後、大規模言語モデルが人々が新しい知識を学ぶのを強力にサポートするようになることが期待されている。

しかし、日本語の分野では、教育を目的とする数学専門の大規模言語モデルはまだ存在していない。そこで本稿では、教育目的のため、初等数学問題を解決し、手順を示すことができる大規模言語モデルの構築方法について報告する。具体的には、まず、解答プロセスを含む日本語初級レベルの数学データセットを構築し、それを利用して、オープンソースの StableLM をベースモデルとして教師ありファインチューニングを行い、数学に特化したモデル JEElementaryMathGPT を構築した。評価実験の結果、7B サイズの JEElementaryMathGPT は 7B サイズと 13B サイズの Vicuna1.5 や 7B サイズと 13B サイズの WizardMath1.0 を超える能力を持つことが明らかになった。

2 背景

近年、オープンソースの事前学習済み大規模言語モデル [11, 15, 16] が増えており、特定のドメインに特化した大規模言語モデルを訓練することが可能になった。

大規模言語モデルの訓練では、まず大量のデータから文の次の単語を予測する方法で基本的な言語モデルを得る [17]。この段階は事前学習 (プリトレーニング) と呼ばれている。次に、研究者は人間

* 共同第一著者

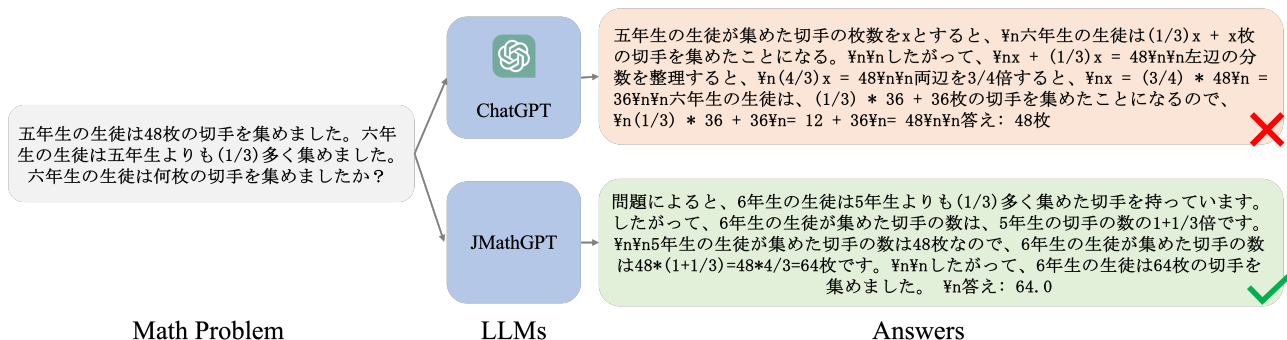


図1 例：ChatGPT (GPT-3.5-Turbo) は非常に簡単な数学問題でさえも、解けないことがある。

の指示に大規模言語モデルを従わせるため、教師ありファインチューニング (SFT) と人間のフィードバックからの強化学習 (RLHF) という二つの手法が用いられる [18]。SFT は大規模言語モデルを人間の指示に一致させるための最初のステップである。このステップでは、質の高い教師データ (人間の指示とモデルが出力すべき返答) を用いて大規模言語モデルを学習させる。

RLHF の段階では、まず大規模言語モデルに同じ指示に対して複数の回答を生成させ、その後、人間からのフィードバックと強化学習を用いて報酬モデルを訓練する。その後、この報酬モデルを使用し、人間の好みの基準に最も合致する答えを大規模モデルに生成させる。RLHF 段階の訓練は非常に煩雑であるため、最近この段階を簡素化することが試みられている。例えば、大規模モデル自体を報酬モデルとして使用することを提案した DPO [19] と呼ばれる手法が存在する。これらの手法を用いることで、人間の指示に従い、与えられた様々なタスクをこなすことができる大規模言語モデルを構築することができる。

3 実験

日本語数学問題に特化した大規模言語モデル JElementaryMathGPT を訓練するため、我々は SFT を使用し、FastChat [20] ライブラリ¹に基づき、JElementaryMathGPT を実装する。

3.1 データセット構築

オープンソースの日本語数学問題データセットは現在存在していないため、まず、推論過程付きの日本語数学問題データセットを構築する。

図2に示すように、中国語の数学問題データセッ

1 <https://github.com/Lm-sys/FastChat>

ト Math23k² (質問と回答のみ) 中の質問と回答を GPT-4 に与えることで、質問に対する推論過程を GPT-4 に生成させる。使うプロンプトは “You will be provided with a math question and its answer in Chinese, and your task is to give the calculation process in Chinese, make sure your answer is the same as the given answer. question[i]” である。次に、推論過程を含む中国語データセットを GPT-3.5-Turbo を使用して日本語に翻訳する。最後に誤った翻訳とエラーを含む推論過程をフィルタリングし、最終的なデータセットとした。訓練セットの問題数が 20445 で、テストセットの問題数が 980 である。

構築した数学データセットを Vicuna のデータセットの対話形式 (ユーザとアシスタント³) に変換して、モデルを学習させる。

3.2 実装の詳細

本研究では Stability AI 社の Japanese StableLM Base Beta 7B モデル⁴をベースモデルとして使用し、訓練を行う。このモデルは、Meta 社の言語モデル Llama-2-7b⁵に基づき、大量の日本語データセットで事前学習を続けていた大規模言語モデルである。ハイパーパラメーターについては、初期学習率は $2e-5$ で、学習率スケジューラのタイプは *cosine* である。0.04 比率でモデルをウォームアップする。

新しく構築した数学データセットでモデルを3エポックの訓練するには、4つの A100 GPU によって勾配累積を16回行いながら、約7時間がかかる。デバイスごとのトレーニングバッチサイズは2である。訓練するとき、アシスタントの出力に対して

2 <https://huggingface.co/datasets/Gxg/Math23k>
 3 https://github.com/Lm-sys/FastChat/blob/main/data/dummy_conversation.json
 4 <https://huggingface.co/stabilityai/japanese-stablelm-base-ja-vocab-beta-7b>
 5 <https://huggingface.co/meta-llama/Llama-2-7b>

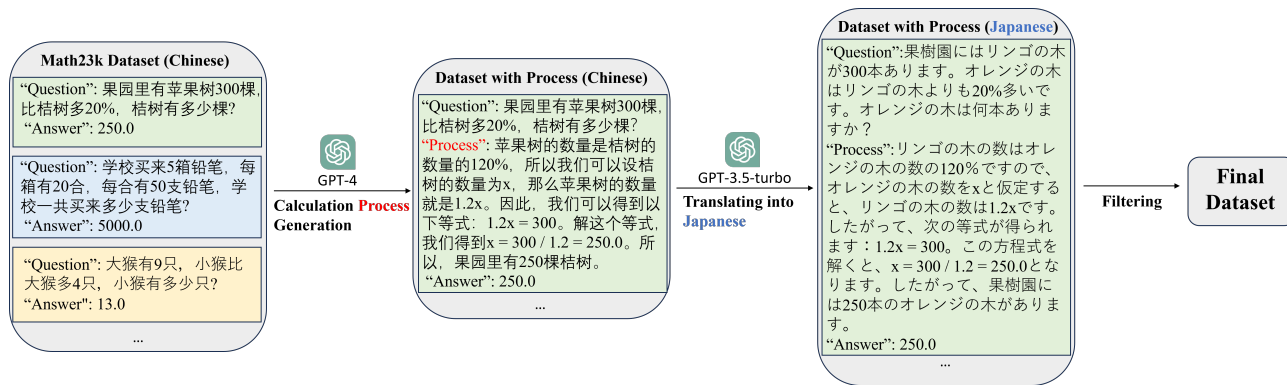


図2 日本語推論過程付きの数学問題データセットの構築手順。

SFT 損失を計算する。

3.3 評価手法

モデルの性能はテストセットで評価し、正解率を報告する。モデルが生成した推論過程に対しては、出力の最後の数字を抽出し、それを浮動小数点数に変換した数値を最終的な答えとする。その後、予測された答えと正しい答えが一定の相対許容差範囲 ($1e-5$) 内にあるかどうかを判断する。

3.4 実験結果

このセクションでは、Vicuna⁶、WizardMath⁷を含むオープンソースの大規模言語モデルと JEElementaryMathGPT を比較する。日本語に対応していない WizardMath に対して、以下のプロンプトを使用することで、WizardMath は英語での推論過程と答えを返し、比較することができる。また、テストセットでの ChatGPT(GPT-3.5-Turbo) の結果も比較する。

ChatGPT のプロンプトは以下のようになる:

以下の質問を答えてください: {problem} 計算過程を示してください、そして最後には次の形式で答えを示してください: '答え:' と記入してください。例: '答え: 0.0'。答えは小数点を含む数字で示してください。

WizardMath のプロンプトは下記の通りになる:

Below is an instruction that describes a task. Write a Japanese response that appropriately completes the request. ### Instruction: {instruction} ### Response: Let's think step by step."

表1に示すように、7B サイズおよび 13B サイズ

のモデルの中で JEElementaryMathGPT が最も良いパフォーマンスを示している。同じく数学問題専門モデルである WizardMath より JEElementaryMathGPT の性能が上回っていることが分かる。

Model Name	Model Size	Accuracy
Vicuna-7B-V1.5	7B	11.35
Vicuna-13B-V1.5	13B	19.90
WizardMath-7B-V1.0	7B	21.33
WizardMath-13B-V1.0	13B	27.55
JEElementaryMathGPT (ours)	7B	35.41
ChatGPT (GPT-3.5-Turbo)	Unknown	51.94

表1 新しく構築されたデータセットのテストセットでモデルの正解率。

3.5 ケーススタディー

本セクションでは、テストセットの中で JEElementaryMathGPT が生成した誤った答えからいくつかのケースに注目し、JEElementaryMathGPT が誤った答えを生成した原因を分析する。

3.5.1 推論過程が正しい、計算ミスなし

以下の例は JEElementaryMathGPT の推論過程が正しく、計算ミスもない例である。この例に対して、ChatGPT(GPT-3.5-Turbo) は正解を生成することができなかった。

Question: “324メートルの道路を修復するため、最初の8時間で240メートルを修復しました。残りの部分は1時間につき21メートル修復します。完了するまでには、あと何時間かかりますか?”

JEElementaryMathGPT: “問題で与えられた道路の総距離は324メートルで、最初の8時間で240メートル修復されました。したがって、残りの距離は324メートルから240メートルを引いたもので、つ

6 <https://huggingface.co/lmsys/vicuna-7b-v1.5>

7 <https://huggingface.co/WizardLM/WizardMath-7B-V1.0>

まり 84 メートルです。

次に、残りの距離を 1 時間あたり 21 メートルずつ修復すると、84 メートルを 21 メートルで割った結果、4 時間かかります。

したがって、残りの距離を修復するには、あと 4 時間かかります。

答え: 4.0”

GPT-3.5-Turbo: “まず、残りの距離を計算します。全体の道路の長さは 324 メートルで、すでに 240 メートル修復されているため、残りの距離は 324 メートル - 240 メートル = 84 メートルです。

次に、1 時間に修復される距離を計算します。1 時間につき 21 メートル修復されるため、残りの距離を修復するのにかかる時間は 84 メートル ÷ 21 メートル/時間 = 4 時間です。

最後に、最初の 8 時間と残りの 4 時間を合計します。合計時間は 8 時間 + 4 時間 = 12 時間です。

答え: 12.0”

正解: “4.0”

この例から、JEElementaryMathGPT が正確な推論過程と答えを生成できて、我々のトレーニングが役に立つことが分かった。

3.5.2 推論過程が正しい、計算ミスあり

以下の例は JEElementaryMathGPT の推論過程が正しいが、計算ミスにより正解にたどり着くことができなかった。一方 ChatGPT は推理及び解答両方に間違いがあった。

Question: “甲城から乙城まで車が 3 時間で 126 キロメートル進みます。この速度で走ると、全行程を終えるのに 7 時間かかります。甲城と乙城の距離は何キロメートルですか?”

JEElementaryMathGPT: “問題によると、車は 3 時間で 126 キロメートル走行できるので、1 時間あたりの走行距離は 126 を 3 で割った値、つまり 42 キロメートルです。したがって、全行程 7 時間の場合、走行距離は 42 を 7 で掛けた値、つまり 284 キロメートルです。したがって、甲乙の 2 つの都市は 284 キロメートル離れています。

答え: 284.0”

GPT-3.5-Turbo: “甲城から乙城までの距離を x キロメートルとします。

甲城から乙城までの距離を車の速度で割ると、車がかかる時間が求められます。この場合は x キロメートル / 126 キロメートル / 時間 = $x / 126$ 時間

です。

また、この速度で走る場合、全行程を終えるまでにかかる時間は 7 時間です。

以上の情報から、以下の方程式が成り立ちます。

$$x / 126 = 7$$

この方程式を解くと、

$$x = 126 * 7$$

$$x = 882$$

よって、甲城と乙城の距離は 882 キロメートルです。

答え: 882”

正解: “294.0”

この例では、JEElementaryMathGPT の推論過程が正しいが、42 と 7 の掛け算を計算する際に間違えて 294 ではなく、284 を生成してしまった。また、大規模言語モデルは数値計算が苦手であることもわかった。

4 今後の課題

外部ツール: テストセットで JEElementaryMathGPT が生成した誤った答えからランダムに抽出した 20 問の中で、生成された答えの誤りの原因を調べてみると、そのうち 16 問は計算ミスを含んでいた。したがって、モデルの算数能力を改善することが結構重要である。次のステップは、計算ミスを減らすために電卓などの外部ツールを使う予定である。

訓練データとモデルのサイズ: 今回学習で使ったデータセットは外国語からの翻訳文で、翻訳文の質による影響があるため、今後、より多く質の高い数学問題データを収集し、より大きいサイズの大規模言語モデルの訓練を試みる予定がある。

5 おわりに

本稿で、教育を目的とする手順を示すことができる日本語初等数学に特化した大規模言語モデルの構築方法について報告する。具体的には、推論過程を含む日本語初等数学問題のデータセットを構築し、オープンソースの StableLM をベースモデルとして学習し、JEElementaryMathGPT を構築した。最終的に、7B サイズの JEElementaryMathGPT が 7B サイズおよび 13B サイズの Vicuna1.5 と WizardMath1.0 よりも優れた性能を持つことが明らかになった。また、JEElementaryMathGPT が誤った答えを生成する原因を分析し、さらにモデルの精度を改善する方法を議論した。

参考文献

- [1] OpenAI. Introducing chatgpt, 2022. <https://openai.com/blog/chatgpt>.
- [2] OpenAI. Gpt-4 technical report, 2023.
- [3] Haoran Xu, Young Jin Kim, Amr Sharaf, and Hany Hassan Awadalla. A paradigm shift in machine translation: Boosting translation performance of large language models, 2023.
- [4] Baptiste Rozière, Jonas Gehring, Fabian Gloeckle, Sten Sootla, Itai Gat, Xiaoqing Ellen Tan, Yossi Adi, Jingyu Liu, Tal Remez, Jérémy Rapin, Artyom Kozhevnikov, Ivan Evtimov, Joanna Bitton, Manish Bhatt, Cristian Canton Ferrer, Aaron Grattafiori, Wenhan Xiong, Alexandre Défossez, Jade Copet, Faisal Azhar, Hugo Touvron, Louis Martin, Nicolas Usunier, Thomas Scialom, and Gabriel Synnaeve. Code llama: Open foundation models for code, 2023.
- [5] Akari Asai, Zeqiu Wu, Yizhong Wang, Avirup Sil, and Hannaneh Hajishirzi. Self-RAG: Learning to retrieve, generate, and critique through self-reflection. **arXiv preprint arXiv:2310.11511**, 2023.
- [6] Tianyi Zhang, Faisal Ladhak, Esin Durmus, Percy Liang, Kathleen McKeown, and Tatsunori B. Hashimoto. Benchmarking large language models for news summarization, 2023.
- [7] Brown et al. Language models are few-shot learners. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, **Advances in Neural Information Processing Systems**, Vol. 33, pp. 1877–1901. Curran Associates, Inc., 2020.
- [8] Chowdhery et al. Palm: Scaling language modeling with pathways, 2022.
- [9] Tiedong Liu and Bryan Kian Hsiang Low. Goat: Fine-tuned llama outperforms gpt-4 on arithmetic tasks, 2023.
- [10] Haipeng Luo, Qingfeng Sun, Can Xu, Pu Zhao, Jianguang Lou, Chongyang Tao, Xiubo Geng, Qingwei Lin, Shifeng Chen, and Dongmei Zhang. Wizardmath: Empowering mathematical reasoning for large language models via reinforced evol-instruct. **arXiv preprint arXiv:2308.09583**, 2023.
- [11] Meta. Llama 2: Open foundation and fine-tuned chat models, 2023.
- [12] Zhangir Azerbayev, Hailey Schoelkopf, Keiran Paster, Marco Dos Santos, Stephen McAleer, Albert Q. Jiang, Jia Deng, Stella Biderman, and Sean Welleck. Llemma: An open language model for mathematics. **arXiv preprint arXiv:2310.06786**, 2023.
- [13] Arto Hellas, Juho Leinonen, Sami Sarsa, Charles Koutchme, Lilja Kujanpää, and Juha Sorva. Exploring the responses of large language models to beginner programmers’ help requests. In **Proceedings of the 2023 ACM Conference on International Computing Education Research - Volume 1**, ICER ’23, p. 93–105, New York, NY, USA, 2023. Association for Computing Machinery.
- [14] Andrew Caines, Luca Benedetto, Shiva Taslimipour, Christopher Davis, Yuan Gao, Øistein E. Andersen, Zheng Yuan, Mark Elliott, Russell Moore, Christopher Bryant, Marek Rei, Helen Yannakoudakis, Andrew Mullooly, Diane Nicholls, and Paula Buttery. On the application of large language models for language teaching and assessment technology. In Steven Moore, John C. Stamper, Richard Tong, Chen Cao, Zitao Liu, Xiangen Hu, Yu Lu, Joleen Liang, Hassan Khosravi, Paul Denny, Anjali Singh, and Chris Brooks, editors, **Proceedings of the Workshop on Empowering Education with LLMs - the Next-Gen Interface and Content Generation 2023 co-located with 24th International Conference on Artificial Intelligence in Education (AIED 2023), Tokyo, Japan, July 7, 2023**, Vol. 3487 of **CEUR Workshop Proceedings**, pp. 173–197. CEUR-WS.org, 2023.
- [15] Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Léo Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. Mistral 7b, 2023.
- [16] Albert Q. Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, Gianna Lengyel, Guillaume Bour, Guillaume Lample, Léo Renard Lavaud, Lucile Saulnier, Marie-Anne Lachaux, Pierre Stock, Sandeep Subramanian, Sophia Yang, Szymon Antoniak, Teven Le Scao, Théophile Gervet, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. Mixtral of experts, 2024.
- [17] Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. Improving language understanding by generative pre-training.
- [18] Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. Training language models to follow instructions with human feedback, 2022.
- [19] Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. In **Thirty-seventh Conference on Neural Information Processing Systems**, 2023.
- [20] Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. Judging llm-as-a-judge with mt-bench and chatbot arena, 2023.