大規模言語モデルにおける 多段推論の依存構造と推論能力の関係検証

榎本倫太郎 [1]　新妻巧朗 [2]　栗田修平 [3]　河原大輔 [1,4]
[1] 早稲田大学大学院　[2] 朝日新聞社 メディア研究開発センター
[3] 国立情報学研究所　[4] 国立情報学研究所 大規模言語モデル研究開発センター
re9484@akane.waseda.jp　niitsuma-t@asahi.jp
dkw@waseda.jp　skurita@nii.ac.jp

概要

大規模言語モデル (LLM) の多段推論能力を測るタスクとして、算術推論や演繹推論タスクがある. 一般にこれらのタスクは必要な推論ステップ数が長い問題ほど解答が難しいとされている. 本研究では推論ステップ数だけでなく、LLM の多段推論間の依存関係に着目し、複雑な依存構造が LLM の最終解答精度にどのように影響するかを分析する. 結果として、問題の提示順序や推論の依存関係の深さ、完全な対称性が正答率に影響を与えることがわかった.

1　はじめに

LLM のテキスト生成能力の向上につれ、その推論能力を測る様々なタスクが提案されている [1, 2, 3]. 特に算術推論や演繹推論タスクは途中計算や、いくつかの前提を利用して結論を出す必要があるなど、最終解答を得るために複数回の推論を要する. 算術推論タスクの例として、 GSM8K [1] や MATH [2] が存在する. GSM8K の問題例を図 1 に示す. これは James が 1 年間に書く手紙のページ数を求める問題である. この問題における推論の途中結果を木構造で表現したものを図 2 に示す. 624 pages が最終解答で、それ以下のノードは最終解答を求めるための途中結果である. 1 年間に書くページ数を求める前に 1 週間あたりのページ数を求める必要があるなど、推論には依存関係が存在する. 一般に、途中過程で必要となる計算ステップ数 (図 2 の非葉ノード数) が増加すると問題が難しくなる傾向がある [4, 5].

GSM8K の問題の解答精度を高めるために、より多様で難易度の高いデータセットを LLM によって生成し、事後学習に用いることがある [4, 6, 7]. その他に、GSM8K の計算過程を木構造に変換し、木を

> James writes a 3-page letter to 2 different friends twice a week. How many pages does he write a year?

図 1　GSM8K の問題例

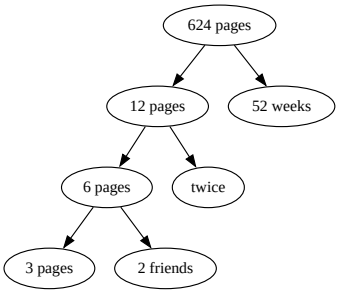

図 2　GSM8K 問題例の推論依存関係

複雑化した後に文章問題表現に戻すことでデータセットを拡張する手法がある [8]. また、多段推論タスクにおける前提の順序付けの影響を示すために、GSM8K の問題文順序を入れ替えた R-GSM が提案されている [9]. しかし、これまでに各推論の依存関係、特に依存構造をもとに LLM の解答精度を詳細に分析した研究は存在しない.

本研究では、多段階の推論タスクにおいて推論に必要なステップ数の他にどのような要素が LLM の解答に影響を与えるのかを調査する. 多段階の推論構造を把握しやすくするために、木構造をもとに算術演算タスク、大小比較タスク、演繹推論タスクの 3 つを定義し、複数の推論の依存構造と解答精度を分析する. 分析の結果、ステップ数や前提順序の他に、依存構造の深さや完全な対称性が解答精度に影響を与えていることがわかった. この研究が今後の多段推論タスクにおけるデータセット拡張に貢献することを期待する.

2 関連研究

推論の依存関係に着目したデータセットやその拡張手法として、Ye らの iGSM [5] や Zhang らの DARG フレームワーク [8] が存在する. iGSM では 4 つの階層的なカテゴリーを事前定義し、グラフ化された依存関係をもとに算術推論データセットを作成している. 限られた依存構造に対して分析を行っているため、依存構造の複雑さに対する詳細な調査ではない. DARG フレームワークでは、既存の GSM8K の問題からより難しい問題を生成するために、問題の木構造に着目している. 具体的には、既存問題における推論の依存構造を LLM によって木で表現し、その木構造をルールベースで複雑化した後に新たな問題文を LLM で生成する. この研究では、依存関係の深さや幅を増やすほど正答率が低下することが示されている. しかし、推論ステップ数の増加による影響と切り離して分析されていないため、依存構造自体の影響であるかは不明である.

一方で問題の依存構造に対してその前提順序をどのように提示するかも重要な要素となる. Chen らは、 Zhang らの論理的推論タスク [10] を改良した演繹推論タスクにおいて前提順序の変更が解答精度に影響を与えることを示している [9]. 前提順序として順方向、逆方向、シャッフルを比較し、順方向以外の提示順ではモデルによって大きく精度が落ちる. しかし、この研究では前提の依存構造自体には焦点を当てていない. また、この前提には推論に必要のない情報も含んでいるため分析が複雑になる.

3 検証データセットの構築

3.1 概要

算術演算、大小比較、演繹推論の 3 種類のタスクのデータセットを構築し、LLM の多段推論能力を検証する. データセットは、木で表現された依存構造パターンをもとに問題を作成する. 図 3 に算術演算タスク（加算のみ）のデータセット構築手法の概要を示す. 大きく分けて木構造の生成、値の割り当て、自然言語への変換の段階が存在する. 図 4 に作成された問題の例を示す. 算術演算タスクでは図 3 における根ノードの数値を最終解答とする. 他のタスクについては付録 A.1 にて補足する. 重要な点は、どのタスクにおいてもある非葉ノードはその子ノードの結果によって推論されるという依存関係である.

表 1 二分木の深さとパターン数

深さ	1	2	3	4	5	6
パターン数	1	2	7	56	2,212	2,595,782
非葉ノード数	1	2-3	3-7	4-15	5-31	6-63

3.2 木構造生成と値の割り振り

依存関係の元となる木構造は、簡単のために非葉ノードが必ず 2 つの子ノードを持つ二分木に限定する. 生成の具体的なアルゴリズムは次の通りである.

1. 根ノードのみを深さ 0 として定義
2. 深さ d の木は、根の左側に深さ d-1 の木集合から、右側に深さ d-1 以下の木集合から全ての組み合わせを選択し配置
3. 木のパターンを再帰的に生成

なお、ある非葉ノードの左部分木と右部分木を入れ替えて同じになる木は同一とみなす. これは、左右を入れ替えても構造自体の複雑さは変化しないためである. 同一なものを考慮した木の種類をパターンと呼ぶ. 深さ 1 から 6 までの二分木を生成し、そのパターン数、非葉ノード数は表 1 のとおりである. 木のパターン数だけ異なる依存構造が存在する. 表 1 の深さ 5、6 ではそのパターン数が多いため、実際の問題作成においては生成された木をサンプリングして使用する[1].

二分木のパターンの生成後、図 3 に示されるように値を割り当てる. まずは全てのノードに A, B, C, ... のようなパラメータ名を割り振り[2]、葉ノードに対して 10, 5 のような値を[3]、非葉ノードに＋のような演算子[4]を割り振る. 非葉ノードの値は、葉ノードの値と自身の持つ演算子をもとに計算し、最後に根ノードの値を求める. これが答えとなる.

3.3 木構造から自然言語表現への変換

値を割り振った二分木から自然言語表現に変換する. 具体的には、図 3 のように各ノード A, B, C, ... を数式表現に書き換える. これをノードの宣言とする.

1) 深さ 1 から 4 までは全ての木に対して問題を 100 個ずつ作成した. 深さ 5（6）ではステップ数ごとに 10（5）種類の木をランダムにサンプリングし、木構造ごとに 20（10）個の問題を作成した.
2) 実際は A, B, C, ... のようなアルファベットを順に割り振るのではなく、ランダムな 2〜3 字のアルファベットを割り振る.
3) 算術演算タスクでは 0〜30、大小比較タスクでは 0〜100 の数値、演繹推論タスクでは True もしくは False が値である.
4) 算術演算タスクでは＋, −, ×を、大小比較タスクでは max, min を、演繹推論タスクでは and, or をランダムに割り振る.

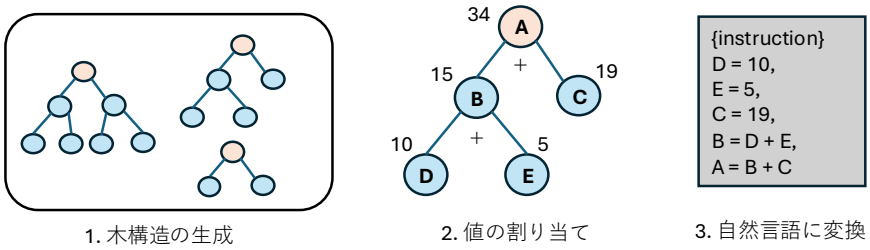

図 3　データセット構築手法（算術演算タスク（加算のみ）の例）

> Problem: Calculate the value of A, where D = 10, E = 5, C = 19, B = D + E, A = B + C.

図 4　算術演算タスク（加算のみ）の問題例

どのタスクにおいてもノードの宣言の順序は自由であり、B = D + E か、A = B + C のどちらを先に宣言しても最終的な A の推論結果は変わらない. 本研究では葉ノードと非葉ノードの宣言を区別し、非葉ノードの宣言順序について分析する. 図 4 のように葉ノードを先（もしくは後）にまとめて宣言し、非葉ノードは根ノードを基準に幅優先とその逆順、深さ優先とその逆順、およびランダム順の 5 つの順序で宣言する. 例えば、図 3 の木において幅優先は A → B → C の順序で宣言する. また、幅優先の逆順は C → B → A の順序となる. どの宣言順序が LLM にとって推論しやすいかを実験を通じて考察する. 以降の実験で用いる式の提示順序はこのノードの宣言順序と同義である.

4　実験

4.1　実験設定

実験には Meta の LLaMA-3.1-8B-Instruct[5] モデルを用いる. 比較対象として、Mistral-7B-Instruct-v0.3[6] モデルと Qwen2.5-Math-7B-Instruct[7] モデルを使用する. また解答の形式を揃え、最終結果が特定しやすいように Greedy search で推論する. 最大出力トークン数は 4,096 とし、この範囲で正解が出ない場合は不正解とする. 推論効率向上のために vLLM [11] を利用する.

5）https://huggingface.co/meta-llama/LLaMA-3.1-8B-Instruct
6）https://huggingface.co/mistralai/Mistral-7B-Instruct-v0.3
7）https://huggingface.co/Qwen/Qwen2.5-Math-7B-Instruct

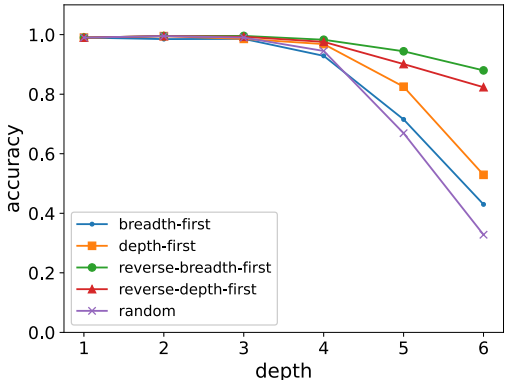

図 5　式の提示順序と正答率（算術演算タスク（加算））: breadth-first は幅優先、depth-first は深さ優先、reverse はそれらの逆順である.

4.2　実験結果

式の提示順序、さらに推論の依存構造に関して、木の深さと対称性に着目し分析する.

4.2.1　式の提示順序の影響

式の提示順ごとに LLaMA-3.1-8B-Instruct モデルの正答率を図 5 に示す. 図より、どの提示順序であっても木が深くなるにつれて正答率が低下していることがわかる. しかし幅優先や深さ優先の逆順の正答率は、深さ 6 の木でも 8 割以上に保たれている. これは式を葉ノードに近い側から、つまり計算可能なノードから提示した方が良いということを示している[8]. 付録 A.2 に示すように、この傾向は演繹推論タスクを除き、他のタスクにおいても同様であった. さらに、付録 A.3 に示す他のモデルの結果でも同様の傾向が見られた. しかし、演繹推論タスクは深さ優先、逆深さ優先の順序の正答率が大きく低下しており、タスクによって解きやすい提示順序は異なる.

また、図 6 に式の各提示順における推論ステップ

8）加算のみに限らず、加算と減算、加算と減算と乗算を含む算術演算タスクでも逆順の正答率は他と比べ高い.

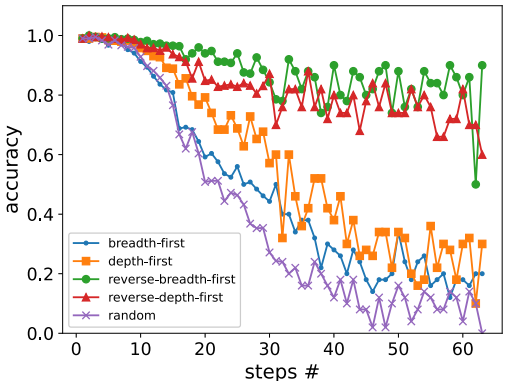

図 6　推論ステップ数と正答率（算術演算タスク（加算））

数 (非葉ノード数) ごとの正答率を示す. 図より全体として推論ステップ数が多いほど、正答率が低下していることがわかる. 図 5 の深さ増加による正答率の低下には推論ステップ数の増加が一つの要因となっていると考えられる.

4.2.2　木構造の深さの影響

次に木構造の深さの影響を考える. 推論ステップ数の影響を排除するため、ステップ数を揃えて分析する. 表 1 の深さごとの非葉ノード数をもとに、各深さで重複する推論ステップ数における正答率を平均することで比較する. 例えば、深さ 3 と 4 の結果を比較するために、重複する 4 から 7 ステップの問題の正答率を平均する. ただし、提示順序の影響を無視できるように式はランダムに提示する.

表 2 は深さ 3 から 6 までの木構造をもとにした問題の正答率を異なる深さ間で比較したものである. Δ は深い木と浅い木のそれぞれの平均正答率を差であり、深さ 3 と 4 の比較だと、96.7%と 98.9%の差が-2.2 となっている. 表 2 より、深さ 4 と 6 の比較を除き、加算、大小比較タスクにおいて同じ推論ステップ数でも依存構造が深い方が正答率が低いことがわかる. これは推論回数のみが多段推論の難しさの要因ではなく、依存構造の深さも関係していることを示す. この傾向は Mistral や Qwen など他のモデルでも同様であった.

4.2.3　木構造の対称性の影響

木構造には根ノードに対して左右同一の部分木を持つものがある. これを対称な木とし、対称性が問題の正答率に影響するかどうかを分析した. ただし、提示順序の影響を無視できるようにランダム順

表 2　二分木の深さと正答率

	正答率 [%]					
	算術演算（加算）			大小比較		
深さ	浅い	深い	Δ	浅い	深い	Δ
3vs4 (step 4-7)	**98.9**	96.7	-2.2	**98.7**	97.7	-1.0
3vs5 (step 5-7)	**98.5**	94.5	-4.0	**98.7**	92.9	-5.8
4vs5 (step 5-15)	**92.9**	88.5	-4.4	**93.9**	93.2	-0.7
4vs6 (step 6-15)	**92.6**	84.8	-7.8	94.0	**94.2**	+0.2
5vs6 (step 6-31)	**64.4**	58.1	-6.3	**83.8**	81.3	-2.5

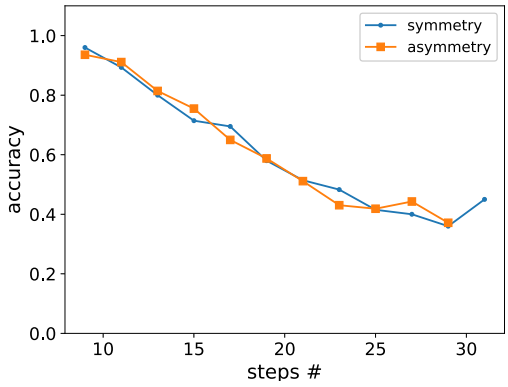

図 7　対称・非対称な木構造問題の正答率（加算タスク）

での提示のみを考えている. 図 7 は深さ 5 の木のうち対称・非対称の木構造を持つ加算問題の正答率をステップ数ごとにプロットしたものである. 図より、各ステップ数で正答率の差があるものの一貫して対称・非対称のいずれかの正答率が高いということはなかった.

対称な木の正答率は、ステップ数が増加するにつれ基本的に単調減少であるが、最後のステップ数 31 の完全二分木をもとにした問題では正答率が大きく向上する. 具体的には、完全二分木の正答率は約 45%と、ステップ数が 6 も小さいステップ数 25 における正答率よりも高い. この傾向は大小比較タスクや他の深さにおいても同様であり、完全二分木をもとにした問題は LLM がその依存構造を認識しやすいのではないかと考える.

5　おわりに

本研究では、推論の依存関係に注目し大規模言語モデルの多段推論能力を調査した. 木構造をもとに問題を作成することで問題の構造的な分析を可能とし、正答率に影響を与える推論ステップ数以外の重要な要素を特定した. 特に式の提示順序や木の深さ、完全な対称性によって問題の難易度が変化することがわかった. 今後はこの知見を利用して新たなデータセット拡張手法の提案をしていきたい.

謝辞

本研究は文部科学省補助事業「生成 AI モデルの透明性・信頼性の確保に向けた研究開発拠点形成」の支援を受けた.

参考文献

[1] Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. Training verifiers to solve math word problems. 2021. abs/2110.14168.

[2] Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the math dataset. 2021. abs/2103.03874.

[3] Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde De Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, et al. Evaluating large language models trained on code. 2021. abs/2107.03374.

[4] Chen Li, Weiqi Wang, Jingcheng Hu, Yixuan Wei, Nanning Zheng, Han Hu, Zheng Zhang, and Houwen Peng. Common 7b language models already possess strong math capabilities. 2024. abs/2403.04706.

[5] Tian Ye, Zicheng Xu, Yuanzhi Li, and Zeyuan Allen-Zhu. Physics of language models: Part 2.1, grade-school math and the hidden reasoning process. 2024. abs/2407.20311.

[6] Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A Smith, Daniel Khashabi, and Hannaneh Hajishirzi. Self-instruct: Aligning language models with self-generated instructions. 2022. abs/2212.10560.

[7] Can Xu, Qingfeng Sun, Kai Zheng, Xiubo Geng, Pu Zhao, Jiazhan Feng, Chongyang Tao, and Daxin Jiang. Wizardlm: Empowering large language models to follow complex instructions. 2023. abs/2304.12244.

[8] Zhehao Zhang, Jiaao Chen, and Diyi Yang. Darg: Dynamic evaluation of large language models via adaptive reasoning graph. 2024. abs/2406.17271.

[9] Xinyun Chen, Ryan A Chi, Xuezhi Wang, and Denny Zhou. Premise order matters in reasoning with large language models. 2024. abs/2402.08939.

[10] Honghua Zhang, Liunian Harold Li, Tao Meng, Kai-Wei Chang, and Guy Van Den Broeck. On the paradox of learning to reason from data. In **Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence**, IJCAI '23, 2023.

[11] Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph Gonzalez, Hao Zhang, and Ion Stoica. Efficient memory management for large language model serving with pagedattention. In **Proceedings of the 29th Symposium on Operating Systems Principles**, pp. 611–626, 2023.

A　付録

A.1　定義したタスクの補足

算術演算タスク以外のタスクについて補足する．次に示す問題例は一部指示文を省略している．図 8 に大小比較タスクの例である．二つの値のうち大きい方か、小さい方かのいずれかを判断し、その値を代入する．最終的に根ノードの数値を推論する．

図 9 に演繹推論タスクの例を示す．ルールと事実が与えられ、事実に対してルールを適用し、Alice が A（実際はランダムな文字列）であるかを推論する．True か False かの二値判定である．なお、演繹推論タスクは Zhang らの論理的推論タスク [10] を参考にした．

> Problem: Based on the given formulas, compare the two numbers . . .
> Answer the value of A.
> Given formulas: D = 10, E = 5, C = 19, B = max(D, E), A = min(B, C).

図 8　大小比較タスクの問題例：max は 2 つの数値の大きい方を min は小さい方を意味する．

> Question: Is Alice A? Sequentially infer information about Alice based on the facts and the rules, and ultimately determine whether Alice is A or not.
> . . .
> Rules: If D and E, then B. If B or C, then A.
> Facts: Alice is not D. Alice is E. Alice is C.

図 9　演繹推論タスクの問題例

A.2　他のタスクの結果

算術演算タスク以外のタスクにおける LLaMA-3.1-8B-Instruct の結果を示す．図 10 は大小比較タスク、図 11 は演繹推論タスクにおける前提の提示順序と正答率の結果である．

A.3　他のモデルの結果

他のモデルの結果を一部ここに示す．図 12 に Mistral-7B-Instruct- v0.3 モデルの大小比較タスクにおける式の提示順序ごとの正答率を示す．図 13 に Qwen2.5-Math-7B-Instruct モデルの算術演算タスク（加算）における式の提示順序ごとの正答率を示す．

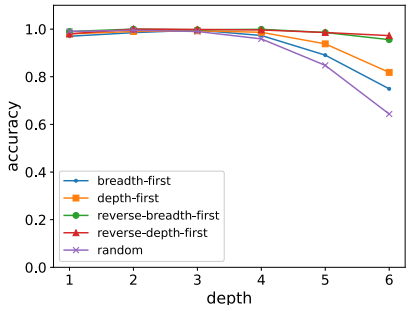

図 10　式の提示順序と正答率（大小比較タスク）

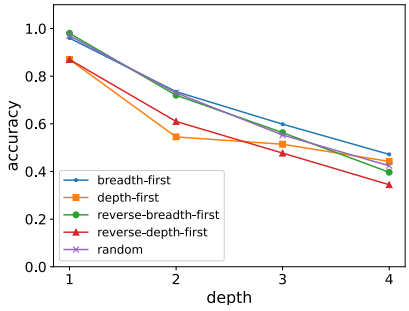

図 11　前提の提示順序と正答率（演繹推論タスク）

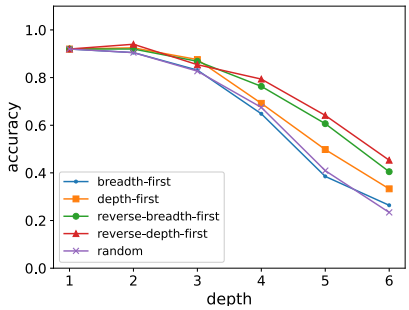

図 12　式の提示順序と正答率（Mistral, 大小比較タスク）

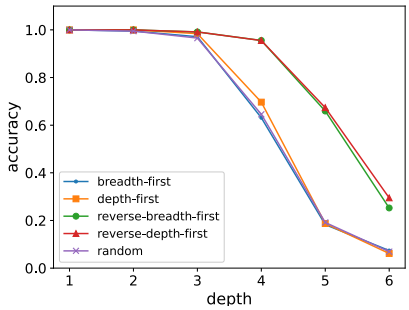

図 13　式の提示順序と正答率（Qwen, 算術演算タスク（加算））