周期表タスクにおける言語モデルの創発的能力と言語モデルが持つ記憶について

星 康人　宮下 大輔　森岡 靖太　Youyang Ng　鳥井 修　出口 淳
キオクシア株式会社
{yasuto1.hoshi, daisuke1.miyashita, yasuhiro.morioka, youyang.ng,
osamu.torii, jun.deguchi}@kioxia.com

概要

近年，大規模な自己回帰言語モデルは広範な下流タスクを追加学習なしで実行できることが判明している．下流タスクにおけるモデルの性能は，一般的にはモデルサイズに対してスケールすることが知られている．しかし，広範な下流タスクにおける性能とモデルサイズとの関係を調べた実験において，モデルサイズの増加に対して不連続な性能改善がみられるタスクの存在が報告されており，このような傾向は創発的能力と呼ばれている．特に知識集約型タスクで創発的能力が現れる理由について，問題に正解するための知識がモデルサイズに対して不連続に偏在するためなのか，あるいはその他の理由があるのか，未だ解明されていない．本研究では，創発的能力が見られる知識集約型タスクの 1 つである周期表タスクについて，モデルが保持する知識とモデルサイズとの関係について調べた．解析の結果，周期表タスクの回答に必要な知識は小規模なモデルも獲得していること，またその知識の正確さはモデルサイズに対しスケールすることが判明した．

1　はじめに

近年，大規模な自己回帰言語モデルは広範な下流タスクを追加学習なしに実行する能力があることが判明している [1, 2]．下流タスクにおける性能とモデルサイズ，すなわちモデルの学習可能パラメータ数との関係を調べた研究において，タスクの精度がモデルサイズの増加に対して不連続に改善するタスクがあることが報告されており，このような傾向は創発的能力 (emergent abilities) と呼ばれている [3]．創発的能力について，その起源や理由については未だ解明されていない．

一般に言語モデルの性能は，その規模にスケールすると考えられている．例えば言語モデルそのものの性能指標（事前学習時の損失値）は，モデルサイズ，学習データ量，および事前学習時の計算量の増加のそれぞれに対して単調に改善するスケール則が成り立つことが知られている [4, 5]．

また，モデルが保持できる知識の量は，モデルサイズにスケールすると考えられている [6, 7, 8]．例えば BIG-bench [2] の qa_wikidata[1)] という質問応答タスクでは，モデルサイズを 2 倍にするごとに精度が約 5.3 ポイント向上するという関係（べき乗則）を見ることができる．このようなべき乗則は言語モデルの他，画像認識など自然言語処理以外の分野も含めて深層学習の多くのタスクでも見られる現象であり [9]，理論的な解析も進みつつある [10, 11]．

それに対し，原子番号から元素名を予測するタスクである周期表タスクは，qa_wikidata と同じく知識を問う質問応答タスクであるが，タスクの精度とモデルサイズとの関係は全く異なる．具体的には，学習可能パラメータ数が 10B 程度以下のモデルでは全く正解できないが，それを超える規模のモデルで突如として精度が向上するという創発的能力が発現することが報告されている [2, 3]．

周期表タスクで創発的能力が見られる理由について，回答に必要な知識が大規模なモデルにのみ偏在しているのか，あるいは他の要因があるのか，未だ切り分けができていない．特に前者に着目すると，創発的能力に関する理解を深めるためには，モデルサイズがある閾値を超えたときに突如として知識が獲得されるようになるといった傾向があるかどうか調べる必要がある．

本稿では周期表タスクを題材とし，これに正解するために必要な知識と，それ以外の要素として回答を導く推論能力の 2 つの要素に分解して考える．特

1） https://github.com/google/BIG-bench/tree/main/bigbench/benchmark_tasks/qa_wikidata

に前者に着目し，モデルサイズに応じて周期表に関する知識量がどのように変化するかを調査することで，創発的能力に関する新たな洞察を与える．

2 関連研究

創発的能力は，モデルサイズの増加に対してある規模まで精度が下がるようなタスク [12] において，モデルをさらに大規模にすると精度が向上するような U 字型の精度変化として見られることもある [13]．このことから，小規模なモデルにおける規模と精度の関係を大規模なモデルに対して単純に外挿することでは，大規模なモデルの性能を予測することができないことが示唆されている．

創発的能力を引き出す要因は，モデルサイズの他にも存在すると考えられている．例えば GPT-3 [1] と PaLM [14] は共に学習可能パラメータ数が 100B を超える大規模言語モデルであるが，PaLM でのみ創発が観測されるケースがある [3]．これらのモデルでは，学習データの品質やアーキテクチャが異なるため，創発的能力の誘因がモデルサイズのみではないことが示唆されている．

創発的能力は，周期表タスクのみならず，算術推論等のタスクでも見られる [1, 15, 16]．このようなタスクが複数存在することは，計 200 を超える多様な下流タスクで構成されるベンチマークである BIG-bench においても確認されている [2]．ただし本稿では，知識集約型タスクの 1 つである周期表タスクに焦点を当てて，モデルが保持する周期表の知識について詳細に解析する．

言語モデルにおける学習データの記憶に関する包括的な調査 [6] において，記憶の正確さはモデルの規模，学習データ内での重複数，およびプロンプトで与えるコンテキストの長さに対してスケールすることが判明している．また，固有表現に関する質問応答では，学習データ内での出現頻度が高い固有表現ほど，それに関する問題の正解率が高くなることが報告されている [17]．本稿では，周期表タスクに関連する学習データの記憶精度とタスクの精度に焦点を当てた解析を行う．

下流タスクにおける事前学習済み言語モデルの性能は，追加学習を行わなくともプロンプトの工夫によってある程度は向上することが知られている．このようなプロンプトの工夫は，プロンプトエンジニアリングと呼ばれる [18]．本稿では，モデルが周期表に関する知識をどの程度保持しているか確かめるために，改良したプロンプトを用いた実験も行う．特に，正解のヒントとなるようにプロンプト内で例示する問題と回答の組を原子番号順にすることで，モデルが保持する周期表の知識の精度を調べる．

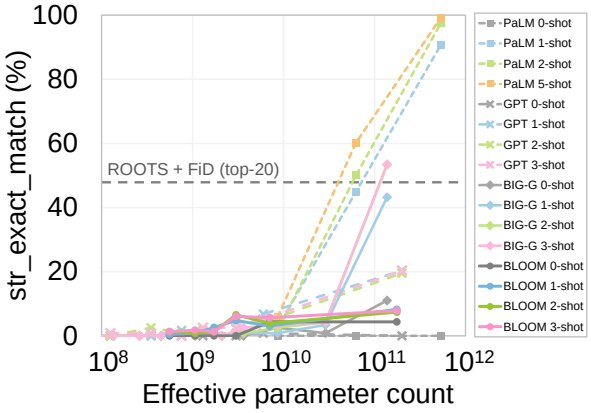

図 1 周期表タスクにおける回答精度．灰色の点線 (ROOTS + FiD) は，ROOTS コーパスから検索した上位 20 位までのパッセージを Fusion-in-Decoder (FiD) の読解器を用いて読解した際の精度 (47.8%) を示す．

3 周期表タスクにおける回答精度

本章では，周期表タスクにおける BLOOM モデルの回答精度を調べる．

3.1 問題設定

実験で使用するデータセットは，BIG-bench [2] に含まれ，周期表に関する知識を問うタスクである periodic_elements タスク[2] うち，原子番号から元素名を予測するタスクである subtask_0（計 23 問）を用いる．本稿ではこのタスクを単に**周期表タスク**と呼ぶことにする．周期表タスクの問題例を下記に示す（下記の問題の答えは “carbon”）．

Q: What is the name of the element with an atomic number of 6?
A:

言語モデルは上記 “A:” 以降に，greedy 探索で得られた系列を出力する．周期表タスクで用いるプロンプトには，無作為に選ばれた問題とその回答の組を例題として，問題文の前に最大 3 つ (3-shot) 付与した系列を使用する．評価方法は先行研究と同様に，モデルがプロンプトに対して greedy 探索で出力した系列の中で最初に現れる元素名と，正解の元素名とを exact match (str_exact_match) で正誤判定する．

2) https://github.com/google/BIG-bench/tree/main/bigbench/benchmark_tasks/periodic_elements

3.2 使用する言語モデル

BLOOM モデル 本稿の実験では，自己回帰言語モデルである BLOOM モデル [19] を使用する．BLOOM モデルは，事前学習済みモデルとしてパラメータ数が 560M から 176B の 6 つのモデルが公開されている[3]．これらのモデルは，計 1.6 TB の多言語コーパスである ROOTS コーパス [20] を用いて事前学習されている．なお，BLOOM モデルの評価結果には，10 回の実験結果の平均値を示す．これは，無作為に選ばれたプロンプト内の例題に依存してモデルの出力が変化し，タスクの精度にばらつきが生じるためである．

ベースライン 周期表タスクにおけるベースラインとして PaLM [14]，GPT [1]，BIG-G [2] の評価結果も用いる[4]．また比較のために，ROOTS コーパスからの検索と読解で得られる精度を示す．ROOTS コーパスからの検索には ROOTS search tool[5] を用いて，0-shot のプロンプトに対する検索上位 20 位までのパッセージを取得する．読解には，Natural Questions [21] データセットで学習された Fusion-in-Decoder [22] の large サイズ (パラメータ数: 738M) の読解器を用いる[6]．

3.3 周期表タスクにおける精度評価

図 1 に，周期表タスクにおける各モデルの回答精度を示す．先行研究で述べられているように，PaLM と BIG-G の回答精度は，パラメータ数が 10B (10^{10}) 程度より大きなモデルにおいて不連続に向上する創発的な傾向が見て取れる．BLOOM モデルの回答精度は，GPT と同様にパラメータ数に対する顕著な精度向上は見られないものの，最大規模のモデルの精度 (3-shot で 7.8%) はそれ以下のモデルと精度比べて高いことが分かる．

また，ROOTS search tool による ROOTS コーパスからの検索と Fusion-in-Decoder の読解器による読解の精度は 47.8%であった．この値は，10B 以下のどのモデルの精度より高いものの，PaLM と BIG-G の大規模なモデルにおける精度を下回っていることが分かる．特に PaLM の 540B のモデルにおける 5-shot の精度 (99.2%) と比較すると，検索と読解による精度は大幅に下回っていることが見て取れる．

4 解析

本章では，ROOTS コーパスに含まれる周期表に関する記述を，BLOOM モデルが出力できるか調べる．また，周期表タスクにおけるプロンプト内の例題を原子番号順で与えた場合の精度を示す．

4.1 BLOOM モデルにおける学習データの記憶精度

BLOOM モデルの事前学習に用いられた ROOTS コーパスから，ROOTS search tool を用いて周期表に関する記述を検索した．付録の表 1 に，その検索結果の代表例を 6 つ示す．この 6 つのパターンを基に，BLOOM モデルにおける周期表に関する学習データの記憶精度を調べた．具体的には，周期表タスクで問われる計 23 の元素に関するプロンプトを作成し，BLOOM モデルの出力を調べた．このとき，プロンプト内で示される例は，原子番号順となっていることに注意されたい．例えば “beryllium” を問うための 3-shot のプロンプトには，表 1 の “...” 以前の文字列が用いられる．

図 2 に，BLOOM モデルにおける学習コーパス内の周期表データの記憶精度に関する評価結果を示す．図より，学習データに含まれる周期表に関する知識の記憶精度は，モデルサイズに対してスケールすること，またショット数が増えるほど正確さが増すことが分かった．以上から，周期表タスクで精度が低い BLOOM モデルでも，回答に必要な知識は獲得していることが分かった．

4.2 プロンプトの改良

前節では，プロンプト内の例を原子番号順に与えることで，BLOOM の小規模モデルでも周期表に関する知識をある程度保持することが判明した．このことから，周期表タスクで与えるプロンプト内の例題を原子番号順にし，学習データと一貫させることで，BLOOM モデルはより正解しやすくなることが予想される．このとき，例えば原子番号 78 番の元素を問う 2-shot のプロンプトは，図 3 のように与えられる．このように，プロンプト内の例題を原子番号順として，周期表タスクを解いた．

3) https://huggingface.co/docs/transformers/model_doc/bloom
4) これらの評価結果には，BIG-bench リポジトリ内で公開されている結果を用いる．https://github.com/google/BIG-bench/tree/main/bigbench/benchmark_tasks/periodic_elements/results
5) BM25 による全文検索を行っている．https://huggingface.co/spaces/bigscience-data/roots-search
6) https://github.com/facebookresearch/FiD

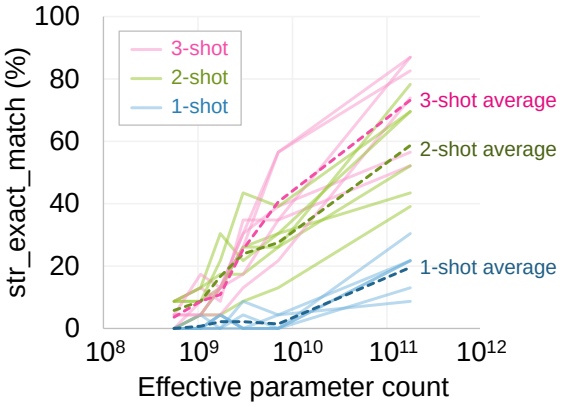

図 2　BLOOM モデルにおける学習データの記憶精度．点線は各 shot における平均の精度を示す．

```
Q: What is the name of the element with an atomic number of 76?
A: osmium
Q: What is the name of the element with an atomic number of 77?
A: iridium
Q: What is the name of the element with an atomic number of 78?
A:
```

図 3　例題を原子番号順とする 2-shot のプロンプトの例．

図 4 に，プロンプト内の例題を無作為に選ぶ場合（図 1 と同様），および原子番号順に与えた際の周期表タスクにおける精度を示す．図より，周期表タスクにおける回答精度はモデルサイズにスケールしており，創発的な変化を示さないことが判明した．また，検索と読解による精度 (47.8%) と比較すると，モデルサイズが 7B のモデルは 3-shot において同等の精度 (47.8%) を達成することが分かった．さらに図 2 と比較すると，学習データに含まれる系列をプロンプトに用いる場合より，図 3 のようなプロンプトを用いる場合の精度が高いことが分かった．

4.3　考察

4.1 節および 4.2 節で得られた結果は，モデルが保持できる記憶の量はモデルサイズに対してスケールするという過去の知見と一貫している．このことから，プロンプト内の例題を無作為に選ぶ subtask_0 の設問方法による評価は，言語モデルが保持する知識の正確さの評価には適さないことが分かる．このことは，subtask_0 の設問方法ではほぼ正答が導けないモデルサイズであっても，周期表に関する知識は正確に保持しているケースが存在することからも見て取れる．すなわち，モデル内の知識の正確さがモデルサイズに対して不連続に変化するということが創発的能力の原因であるという仮説は，棄却されることが示唆される．

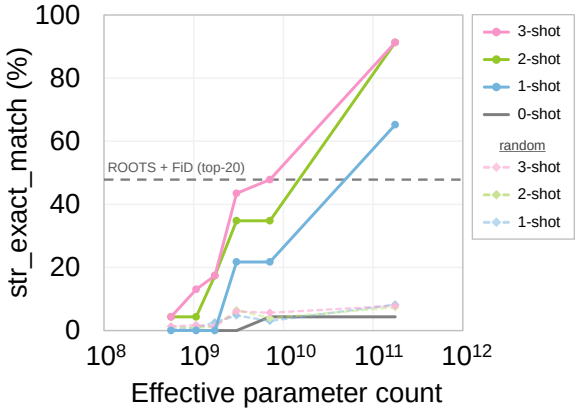

図 4　周期表タスクにおいて，プロンプト内の例題をランダムに与えた場合（点線），および周期表に出現する順番に与えた場合（実線）の精度．灰色の点線 (ROOTS + FiD) は，ROOTS コーパスから検索した上位 20 位までのパッセージを Fusion-in-Decoder (FiD) の読解器を用いて読解した際の精度 (47.8%) を示す．

前記の考察を踏まえ，周期表タスクの創発的能力について再考する．創発的能力が顕著にみられる PaLM モデルに着目すると，PaLM モデルの最大規模のモデルであっても 0-shot では正答できないことが分かる．このことを加味すると，ランダムな順番で与えられた例題とモデル内の知識の中から正答を導く推論能力があるかどうかが，創発前後の差であることが示唆される．その一方で，その推論能力がなぜ不連続な変化をするのかは依然として不明である．その他にも，なぜ知識問題である周期表タスクで創発的能力が見られるか，算術演算等の論理的推論タスクでなぜ創発的能力が見られるかなど，今後の研究で解明すべき課題が存在する．

5　おわりに

本稿では周期表タスクを用いて，言語モデルサイズに対する周期表知識の記憶精度について調べた．その結果，周期表の原子番号と元素名に関する知識の記憶精度は，モデルサイズにスケールすることが分かった．また，周期表タスクを数ショットで実行する際のプロンプト内の例題を原子番号順で与えて学習データと一貫させることで，周期表タスクの回答精度はモデルサイズにスケールすることが判明した．これらの結果から，周期表タスクにおける創発的能力の原因として，周期表に関する知識が小規模なモデル内に存在しないという予想は棄却され，モデル内の知識の中から正しい出力を導く推論能力が創発的であることが示唆される．

参考文献

[1] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. **Advances in neural information processing systems**, Vol. 33, pp. 1877–1901, 2020.

[2] Aarohi Srivastava, Abhinav Rastogi, Abhishek Rao, Abu Awal Md Shoeb, Abubakar Abid, Adam Fisch, Adam R Brown, Adam Santoro, Aditya Gupta, Adrià Garriga-Alonso, et al. Beyond the imitation game: Quantifying and extrapolating the capabilities of language models. **arXiv preprint arXiv:2206.04615**, 2022.

[3] 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. Emergent abilities of large language models. **Transactions on Machine Learning Research**, 2022.

[4] Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models. **arXiv preprint arXiv:2001.08361**, 2020.

[5] Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, and et al. Training compute-optimal large language models. **arXiv preprint arXiv:2203.15556**, 2022.

[6] Nicholas Carlini, Daphne Ippolito, Matthew Jagielski, Katherine Lee, Florian Tramer, and Chiyuan Zhang. Quantifying memorization across neural language models. **arXiv preprint arXiv:2202.07646**, 2022.

[7] Benjamin Heinzerling and Kentaro Inui. Language models as knowledge bases: On entity representations, storage capacity, and paraphrased queries. In **Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume**, pp. 1772–1791, Online, April 2021. Association for Computational Linguistics.

[8] Adam Roberts, Colin Raffel, and Noam Shazeer. How much knowledge can you pack into the parameters of a language model? In **Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)**, pp. 5418–5426, Online, November 2020. Association for Computational Linguistics.

[9] Tom Henighan, Jared Kaplan, Mor Katz, Mark Chen, Christopher Hesse, Jacob Jackson, Heewoo Jun, Tom B Brown, Prafulla Dhariwal, Scott Gray, et al. Scaling laws for autoregressive generative modeling. **arXiv preprint arXiv:2010.14701**, 2020.

[10] Yasaman Bahri, Ethan Dyer, Jared Kaplan, Jaehoon Lee, and Utkarsh Sharma. Explaining neural scaling laws. **arXiv preprint arXiv:2102.06701**, 2021.

[11] Yi Tay, Mostafa Dehghani, Jinfeng Rao, William Fedus, Samira Abnar, Hyung Won Chung, Sharan Narang, Dani Yogatama, Ashish Vaswani, and Donald Metzler. Scale efficiently: Insights from pretraining and finetuning transformers. In **International Conference on Learning Representations**, 2022.

[12] Ian McKenzie, Alexander Lyzhov, Alicia Parrish, Ameya Prabhu, Aaron Mueller, Najoung Kim, Sam Bowman, and Ethan Perez. Announcing the inverse scaling prize ($250k prize pool), 2022. https://www.lesswrong.com/posts/eqxqgFxymP8hXDTt5/announcing-the-inverse-scaling-prize-usd250k-prize-pool.

[13] Jason Wei, Yi Tay, and Quoc V. Le. Inverse scaling can become u-shaped. **arXiv preprint arXiv:2211.02011**, 2022.

[14] Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. Palm: Scaling language modeling with pathways. **arXiv preprint arXiv:2204.02311**, 2022.

[15] Deep Ganguli, Danny Hernandez, Liane Lovitt, Amanda Askell, Yuntao Bai, Anna Chen, Tom Conerly, Nova Dassarma, Dawn Drain, Nelson Elhage, et al. Predictability and surprise in large generative models. In **2022 ACM Conference on Fairness, Accountability, and Transparency**, pp. 1747–1764, 2022.

[16] Alessandro Stolfo, Zhijing Jin, Kumar Shridhar, Bernhard Schölkopf, and Mrinmaya Sachan. A causal framework to quantify the robustness of mathematical reasoning with language models. **36th Conference on Neural Information Processing Systems (NeurIPS 2022) Workshop on Math-AI**, 2022.

[17] Alex Mallen, Akari Asai, Victor Zhong, Rajarshi Das, Hannaneh Hajishirzi, and Daniel Khashabi. When not to trust language models: Investigating effectiveness and limitations of parametric and non-parametric memories. **arXiv preprint arXiv:2212.10511**, 2022.

[18] Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. **arXiv preprint arXiv:2107.13586**, 2021.

[19] Teven Le Scao, Angela Fan, Christopher Akiki, Ellie Pavlick, Suzana Ilić, Daniel Hesslow, Roman Castagné, Alexandra Sasha Luccioni, François Yvon, Matthias Gallé, et al. Bloom: A 176b-parameter open-access multilingual language model. **arXiv preprint arXiv:2211.05100**, 2022.

[20] Hugo Laurençon, Lucile Saulnier, Thomas Wang, Christopher Akiki, Albert Villanova del Moral, Teven Le Scao, Leandro Von Werra, et al. The bigscience roots corpus: A 1.6 tb composite multilingual dataset. In **Thirty-sixth Conference on Neural Information Processing Systems Datasets and Benchmarks Track**.

[21] Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, and et al. Natural questions: A benchmark for question answering research. **Transactions of the Association for Computational Linguistics**, Vol. 7, pp. 452–466, 2019.

[22] Gautier Izacard and Édouard Grave. Leveraging passage retrieval with generative models for open domain question answering. In **Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume**, pp. 874–880, 2021.

A　付録：学習データ内の周期表に関する記述の例

BLOOM モデルの事前学習に用いられた ROOTS コーパスから，ROOTS search tool を用いて周期表に関する記述を検索したところ，代表的な 6 つのパターンとして表 1 に示すような記述がみられた．

表 1　ROOTS コーパスに含まれる，周期表の元素に関する記述の例．

hydrogen, helium, lithium, ...
hydrogen (1), helium (2), lithium (3), ...
1-Hydrogen (H) 2-Helium (He) 3-Lithium (Li) 4 ...
1 H Hydrogen 2 He Helium 3 Li Lithium 4...
1 Hydrogen H · 2 Helium He · 3 Lithium Li · 4 ...
hydrogen (H, atomic number 1); helium (He, atomic number 2); lithium (Li, atomic number 3); ...