算術問題におけるニューラルモデルの構成的推論能力

工藤慧音 [1]　青木洋一 [1]　栗林樹生 [1,2]　Ana Brassard[3,1]
吉川将司 [1]　坂口慶祐 [1,3]　乾健太郎 [1,3]
[1] 東北大学　[2]Langsmith 株式会社　[3] 理化学研究所
{keito.kudo.q4, youichi.aoki.p2 }@dc.tohoku.ac.jp
{kuribayashi, yoshikawa, keisuke.sakaguchi, kentaro.inui}@tohoku.ac.jp
ana.brassard@riken.jp

概要

未知の問題への汎化を達成する上で，問題の構成要素を捉え既知の知識を組み合わせる能力が重要となる．しかし，近年の言語モデルがどの程度構成性を捉えられているかは記号推論の文脈ではまだ明らかでない．そこで本研究は，多段算術記号推論データセットを用いた実験を行い，構成的推論能力の検証を行なう．具体的には，問題の複雑さを整理したスキルツリーを定義し統制的に分析する．実験の結果，言語モデルは体系性の習得が最も困難であり，単純な数式の組み合わせに対する汎化でさえ難しかった．また追加の分析から，入力系列に書かれない知識へのアクセスが必要な問題で，体系性に対する汎化が特に困難である明らかとなった．

1　はじめに

近年，ニューラルモデルに関する技術の発展により機械による自然言語理解は大きく進歩し，多くのタスクがニューラルモデルによって解決されるようになってきた．しかし，言語処理を行う上で必要な要素の 1 つである記号推論能力のニューラルモデルとの融合については，重要な目標の 1 つとして [1, 2] 調査が進行中である [3]．現状，ニューラルモデルがどの程記号推論を実現については相反する結果が報告されている．例えば，ニューラルモデルが多段推論をある程度解くことができると報告する研究もある一方 [4]，単純な記号演算を行うことさえ困難であるとするものも存在し [5]，どのような観点・条件において，どの程度の推論が実現可能なのかといった統制的な分析が必要とされている．

本研究では，多段算術記号推論データセットを用い，問題の複雑さを制御しながらニューラルモデルの多段推論能力を評価する．特に，未知の問題への汎化の際に重要となる構成性に焦点を当てて分析を行う．具体的には，図 1 に示すように，(i) 体系性，(ii) 生産性，(iii) 代入可能性という 3 つの構成性を軸として調査を行い，単純な問題設定で学習したモデルが，そこで学習した要素を組み合わせ一段複雑な問題を解けるかという，言語モデルの構成的な汎化性能を評価する．

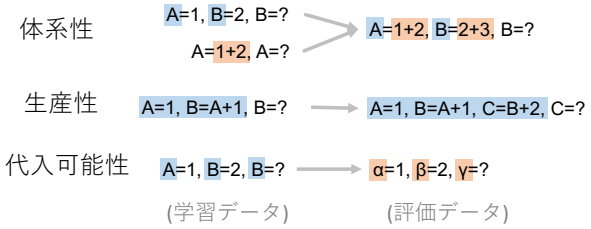

図 1　算術記号推論における構成性の種類

モデルの能力を系統的に調査するために，図 2 に示すように，算術記号推論の複雑さを階層的に定義した構成性に関する**スキルツリー**を導入する．(スキルツリーは教育学の分野において，段階的な学習過程を可視化するための手法を示す用語である [6]．) この階層を足がかりに，ニューラル言語モデルの構成的なの限界を明らかにする．主な実験結果は以下の通りである．

- 3 種類の構成性の中で**体系性**への汎化が最も困難であり，比較的単純な組み合わせであっても汎化を達成することが困難である．
- 特に，入力系列中に明示されていない知識を要する問題において体系性に対する汎化への困難度が増す．
- 本実験の範囲では，途中の推論過程をモデルに学習させた場合であっても，モデルが体系性を捉えることは困難であった．

2　問題設定

典型的な記号推論（手続き型プログラミング言語，アセンブリ言語など）は，少なくとも 3 つの基本的な記号操作，代入 (a=2)，算術演算 (1+2)，参照 (a=?) で構成される．そこで，これら 3 つの基本演算を組み合わせた以下のような多段算術推論問題を解くことを目標とする．

問題： A=1，B=2，C=A+2，C=? **答え：** 3

最後に特定の変数に代入されている値を問われる．答えは必ず一意に定まり，問題設定によっては，最後の問いに直接関わらない数式が含まれる．また，問題は以下の 5 種類の基本的な数式を並べ合わせることで生成される：(i) A=1（代入），(ii) A=B（参照&代入），(iii) A=1+2（演算&代入），(iv) A=B+2（演算&代入&参照，参照），(v) A=?（参照）．

このような算術推論データセットでは，問題文中の数や推論の複雑さを自由に制御することが可能であり，統制した実験を行うために適している．自然言語でのモデルの推論能力評価という視点からは，自然言語の表現の多様さといった難しさとは切り分けて，純粋な算術推論能力を調査する試みと見ることができる．また，本研究で考える算術推論の問題は，DROP [7] に見られるような自然言語の算術推論問題に自然に変換することが可能である．

3　算術推論におけるスキルツリー

3.1　多段記号推論における構成性

構成的な汎化を体系性，生産性，代入可能性の 3 つの軸に分解する [8]．これらの軸を基準として，ニューラル言語モデルがどの程度構成に対する汎化を達成できているかを評価する．

体系性 (systematicity) は，既知の異なる概念を組み合わせてより複雑な概念を理解する能力である．この能力を評価するため，初めに，数種類の基本的な処理（例: 加算 A=1+2,A=? や選択参照 A=1,B=2,B=?) の学習を行う．その後，学習した基本的な処理の組み合わによって構成される問題（例：A=1+2,B=2+3,B=?) における性能を測定する．

生産性 (productivity) とは既知の短い問題の解き方をもとにして，より長い問題や複雑な問題を理解する能力のことである．この能力を評価するため，初めに，短い数式（例:A=1+2,B=2+3,B=?) でモデルの学習を行う．その後，同様の問題形式のより長い問題

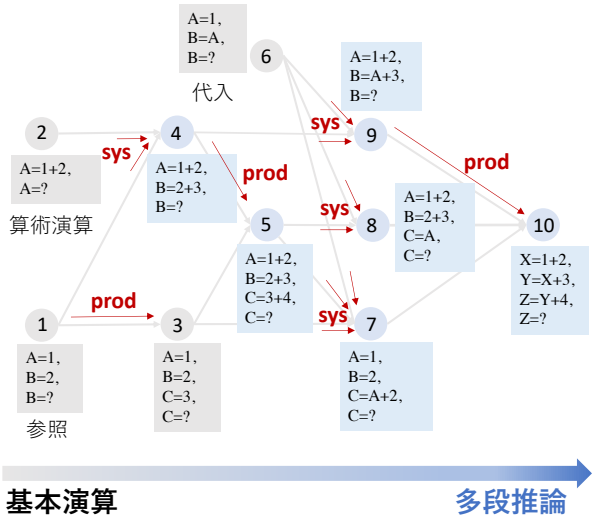

図 2　構成的な汎化能力を評価するためのスキルツリー．事前学習に利用した基本演算は灰色，WA を計算するためのファインチューニング用いたデータは水色で示している．各辺に示した “sys” (= 体系性) “prod” (= 生産性) はそれぞれ，増加する複雑さの種類を表している．

(例： A=1+2,B=2+3,C=3+4,C=?) を解いたときの性能を測定する．

代入可能性 (substitutivity) は，問題中の特定の構成要素が他の (見たことがない) 構成要素に置き換わった場合であっても，変わらず推論を行うことができる能力である．この能力を評価するため，学習時に見たことがない語彙が問題文中に現れた時に，ニューラルモデルは推論することができるかどうかを測定する．具体的には，学習時とテスト時で問題文に用いる変数名を変えて実験を行う（例：A=1+2,A=? で学習を行い，その後 α=1+2，α=? を解くことができるかを評価する）．

3.2　スキルツリー

記号推論の複雑さを （階層的に） 整理した構成性に関するスキルツリーを導入し，組み合わせの複雑さが異なる問題を用いて段階的にモデルを評価することで，多段記号推論におけるニューラル言語モデルの弱点を正確に明らかにする．

具体的には，10 種類の記号推論タスクを設計する．各設定の概要と設定間の階層的な関係を図 2 のスキルツリーに示す．各頂点が複雑さの異なるタスク設定に対応し，各辺が階層的な複雑さの増加を表す．各設定のデータについては，図 2 に記述された例題と同じ式の形式で問題を作成しており，変数 (A，B など) と数字 (1，2 など) のみ無作為に変更してデータセットを作成する （詳細は Appendix C を

表 1　それぞれの設定における正解率を示している．“task” の行は各設定におけるドメイン (学習 → 評価) を表している．それぞれのタスク設定はスキルツリー (図 2) の値に対応している．“type” の行はそれぞれの設定において評価の対象としている構成性の種類 (“sys.” = 体系性 “prod.” = 生産性，“subst.” = 代入可能性) を表している．

		base		large		x-large	
Task	Type	ZA	WA	ZA	WA	ZA	WA
1,2	sys.	42.0	83.5	32.0	89.2	53.9	97.2
→4	+subst.	38.3	81.9	30.8	88.7	53.5	97.0
2,3	sys.	34.2	73.9	34.5	84.9	38.7	94.9
→5	+subst.	34.5	76.5	33.3	86.5	41.2	95.9
2,3,6	sys.	38.7	76.1	44.8	87.7	36.1	93.0
→8	+subst.	37.3	74.5	42.8	85.6	35.0	94.1
2,3,6	sys.	56.4	78.6	48.8	82.6	55.2	93.8
→7	+subst.	57.2	78.6	50.0	84.9	53.2	94.7
1,2,6	sys.	24.3	28.4	23.0	28.5	27.3	31.9
→9	+subst.	26.5	28.2	24.3	31.3	28.5	34.2
7,8	sys.	22.8	20.9	26.2	26.4	22.8	26.9
→10	+subst.	21.9	20.7	24.5	26.1	22.7	29.9
1	prob.	100.0	100.0	100.0	100.0	100.0	100.0
→3	+subst.	100.0	100.0	100.0	100.0	100.0	100.0
4	prob.	100.0	99.9	100.0	99.9	100.0	100.0
→5	+subst.	99.9	99.9	100.0	100.0	100.0	99.9
9	prob.	58.4	60.0	62.0	64.7	59.0	62.9
→10	+subst.	59.8	61.4	62.8	65.6	58.3	63.2

参照）．いずれも，一連の記号推論の後に問われた変数に格納された数値を出力する問題である．

4　実験

4.1　実験設定

学習とテストの設定の組み合わせを適切に選択することにより（例えば，図 2 のタスク 1，2 を学習ドメイン，4 を評価ドメイン），モデルの構成的な汎化能力を多面的に評価した．例えば，タスク 1 (A=1+2,A=?) と 2 (A=1,B=2,B=?) を学習データ，タスク 4 (A=1+2,B=2+3,B=?) を評価データとして，演算操作 (a+b) と参照操作 (A=c,B=d,B=?) に対してモデルがどれだけ汎化しているのかを評価できる．

形式的には，$\mathcal{D}_{\text{train}} = \{d_{\text{train}1}, \cdots d_{\text{train}k}\}$ と d_{test} がそれぞれ学習ドメイン集合と評価ドメインを表すものとする．ここで，「ドメイン」とは，スキルツリー (図 2) の特定の頂点に相当する．

学習設定: 学習用ドメイン d_{train} の学習データと，各学習ドメインの問題を解くのに要する基本演算 (代入，算術演算，参照) によって構成された学習データの和集合を用いてモデルを学習する．5 エポック続けて検証データセットにおける正解率が上がらない，または正解率が 100%に到達した時点で学習を停止する．

評価設定: テストドメイン d_{test} のテストデータに対する正解率を計算する．ここでは，学習の効率を測るために，(i) ゼロショット正解率 (ZA) と (ii) 加重平均正解率 (WA) という 2 つの指標を用いた [9]．WA の測定では，テストドメイン d_{test} の学習データを用いてモデルを追加で学習し，モデルのパラメータを更新する度に検証データにおける正解率を測定し加重平均を算出した（詳細は Appendix A を参照のこと）．

モデルと事前学習: 事前学習済みの系列変換モデルとして広く使われている T5 [10] の 3 種類のサイズ（base，large，xl）を使用した．実験は事前学習したパラメータを初期値として行った．これは，最終的な目標は自然言語上での数量推移論 (記号推論) 問題を解くことにあるためである．

4.2　実験結果

表 1 の 1 列目に示した 9 つの学習ドメインと評価ドメインの組み合わせについて実験を行った（学習ドメイン → 評価ドメイン）．6 つの設定については体系性に対して，残り 3 つの設定は生産性に対する汎化性能を評価した．さらに，各設定において，訓練ドメインで用いた変数名とは異なる変数名（例えば，A の代わりに α）を用いた評価データを用いて，代入可能性に対する汎化性能の評価を行った．

表 1 に実験結果を示した．実験の結果以下の 4 つの傾向が見らた．

- 体系性に対する汎化は生産性に対する汎化に比べて困難であった．
- 単純な基本演算の組み合わせ (例: 1,2→4) であっても，少数の学習データからの汎化は困難であった．
- 代入可能性に対する汎化は達成できた．
- モデルサイズによって性能に (特に ZA について) 大きな差は見られなかった．

また，意味解析の文脈でニューラルモデルが体系性に対する汎化能力を欠くことが示唆されていた [11]．今回の結果は算術多段推論の文脈における，この知見を裏付けるものでもある．ここで，体系性に対する汎化を検証する設定（2, 3 → 5) に着目し，

表 2　設定 2，3→5 における，アブレーション実験．“String” は算術演算の代わりに文字列演算を用いた場合の結果である．“Step” は途中の推論過程を生成させた場合の結果である．

	base		large		x-large	
設定	ZA	WA	ZA	WA	ZA	WA
2,3→5	34.2	73.9	34.5	84.9	38.7	94.9
String	37.3	94.6	66.1	98.5	86.9	99.0
Steps	25.8	77.2	33.1	87.1	31.0	95.7

ニューラルモデルがなぜこの設定において苦戦するのかを理解するため，この設定における複雑さを分解し，追加のモデルの性能の分析を行った．

体系性に対する汎化の難しさは算術演算に起因するのか?　四則演算を文字列演算 (詳細は Appendix C を参照のこと) に置き換え，その他は同様の設定で実験を行った．算術演算と文字列演算の違いは，文字列演算は入力系列の要素をコピーするだけで実現できるのに対し（例：`12+34=1234`），算術演算（例：`1+2=3`）はモデルの内部に格納されている算術知識にアクセスする必要がある点である．

実験の結果，モデルサイズが大きい場合には，文字列演算では体系性の弱点を克服する傾向が見られた．（例：x-large モデルでのゼロショット精度で 86.9%）．このことから，入力には示されていない推論中に生成される中間情報へのアクセスが，体系性に対する汎化の難しさの要因となっていると考えられる．

途中の推論過程を見せることに効果はあるのか?　先行研究において，途中の推論過程をモデルに出力させることで，ニューラルモデルの多段推論能力が向上することが示唆されている [12, 13]．同様に途中の推論過程を明示的に生成させることで，先の分析で明らかとなった体系性に対する汎化の難しさが緩和されるかを検証した．具体的には，学習（事前学習と WA を測定するための学習）時に途中の推論過程 (`A=1+2,B=2+3,C=4+5`，`B=?;B=2+3,B=5`) を出力するようにモデルを学習させた．正解率は，途中の推論過程を含む出力が完全に一致する割合とした．実験の結果，途中の推論過程の生成による大きな性能向上は見られなかった（表 2）．このことから，少なくともこのような統制された実験環境においては，途中の推論過程を生成させる手法の効果は限定的であることが明らかとなった．

5　関連研究

ニューラルモデルの構成的な汎化能力の解析と算術多段推論問題については，個別に研究がなされてきた．本研究はこれら 2 つの研究の方向性を統合したものである．ニューラルモデルの構成的な汎化能力については，SCAN [3]，COGS [11]，CFQ [14] などのデータセットを用いて分析が行われてきた．これらは主に意味解析の文脈での構成性に着目しており，記号推論における構成的な汎化能力には着目していない．算術推論に関しては，DROP [7] などのベンチマークを用いてニューラルモデルの能力を分析することが一般的である．しかし，最近このようなデータセットにはタスクを解くための表面的な手がかりがあることが報告されている [15]．そのため，ニューラルモデルが実際にどの程度の算術推論を実現しているか不明であった．統制されたデータセットを用いた本研究は，算術推論でのニューラルモデルの弱点を正確に把握することに寄与している．

6　おわりに

本研究では，構成的な汎化能力という切り口で，近年のニューラル言語モデルの算術多段推論能力の分析を行った．ニューラルモデルの能力を体系的に分析するため，多段記号推論データセットの複雑さを整理したスキルツリーを定義した．実験の結果，ニューラルモデルは体系性に対する汎化が弱点であり，単純な構成要素を組み合わせに対する汎化であっても達成できないことが明らかとなった．また，追加の実験を通して，入力系列に書かれていないモデルに格納された知識にアクセスする際に，体系性に対する汎化が特に困難となることが明らかとなった．さらに，途中の推論過程も生成させるように学習したモデルであっても，体系性を捉えることは困難であった．

今後は，自然言語上での記号推論実現を目指す．そのために，本研究で明らかとなったような言語モデルに不足した記号推論能力を，自由に設計できる特徴がある形式言語で事前学習を行うことで獲得させることを目指す．その後，獲得した能力を後段の自然言語タスクを解くために転移させるという枠組みを想定している．

謝辞

本研究は JST CREST JPMJCR20D2 及び JSPS 科研費 JP22H00524，21K21343 の助成を受けたものです．また，本研究を進めるにあたり多くの協力を賜りました Tohoku NLP グループの皆様に感謝申し上げます．

参考文献

[1] Gary F Marcus. **The algebraic mind: Integrating connectionism and cognitive science**. MIT press, 2003.

[2] A Garcez and L Lamb. Neurosymbolic AI: The 3rd wave. **CoRR**, 2020.

[3] Brenden M. Lake and Marco Baroni. Generalization without systematicity: On the compositional skills of sequence-to-sequence recurrent networks. In **ICML**, 2018.

[4] Peter Clark, Oyvind Tafjord, and Kyle Richardson. Transformers as soft reasoners over language. In Christian Bessiere, editor, **IJCAI**, pp. 3882–3890. International Joint Conferences on Artificial Intelligence Organization, 7 2020. Main track.

[5] Jing Qian, Hong Wang, Zekun Li, Shiyang Li, and Xifeng Yan. Limitations of language models in arithmetic and symbolic induction. **CoRR**, Vol. abs/2208.05051, , 2022.

[6] Gustavo Tondello and Lennart Nacke. A pilot study of a digital skill tree in gameful education. 10 2019.

[7] Dheeru Dua, Yizhong Wang, Pradeep Dasigi, Gabriel Stanovsky, Sameer Singh, and Matt Gardner. DROP: A reading comprehension benchmark requiring discrete reasoning over paragraphs. In **NAACL**, pp. 2368–2378, Minneapolis, Minnesota, June 2019. ACL.

[8] Dieuwke Hupkes, Verna Dankers, Mathijs Mul, and Elia Bruni. Compositionality decomposed: How do neural networks generalise? (extended abstract). In Christian Bessiere, editor, **IJCAI**, pp. 5065–5069. IJCAI, 7 2020. Journal track.

[9] Alon Talmor, Yanai Elazar, Yoav Goldberg, and Jonathan Berant. oLMpics-On What Language Model Pre-training Captures. **TACL**, Vol. 8, pp. 743–758, 12 2020.

[10] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. **JMLR**, Vol. 21, No. 140, pp. 1–67, 2020.

[11] Najoung Kim and Tal Linzen. COGS: A compositional generalization challenge based on semantic interpretation. In **EMNLP**, pp. 9087–9105, Online, November 2020. Association for Computational Linguistics.

[12] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Ed H. Chi, Quoc Le, and Denny Zhou. Chain of thought prompting elicits reasoning in large language models. **CoRR**, Vol. abs/2201.11903, , 2022.

[13] Maxwell Nye, Anders Johan Andreassen, Guy Gur-Ari, Henryk Michalewski, Jacob Austin, David Bieber, David Dohan, Aitor Lewkowycz, Maarten Bosma, David Luan, Charles Sutton, and Augustus Odena. Show your work: Scratchpads for intermediate computation with language models. In **DL4C**, 2022.

[14] Daniel Keysers, Nathanael Schärli, Nathan Scales, Hylke Buisman, Daniel Furrer, Sergii Kashubin, Nikola Momchev, Danila Sinopalnikov, Lukasz Stafiniak, Tibor Tihon, Dmitry Tsarkov, Xiao Wang, Marc van Zee, and Olivier Bousquet. Measuring compositional generalization: A comprehensive method on realistic data. In **ICLR**, 2020.

[15] Hadeel Al-Negheimish, Pranava Madhyastha, and Alessandra Russo. Numerical reasoning in machine reading comprehension tasks: are we there yet? In **EMNLP**, pp. 9643–9649, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics.

[16] Noam Shazeer and Mitchell Stern. Adafactor: Adaptive learning rates with sublinear memory cost. **CoRR**, Vol. abs/1804.04235, , 2018.

表 3　文字列演算の詳細.

演算子名	説明	例
join	文字列の結合	12 + 34 = 1234
reverseJoin	文字列の結合と反転	123 ^ 78 = 87321
strSub	重複する文字の削除. 全ての文字が削除された場合は 0 を返す.	7873 - 73 = 87
stackJoin	引数の文字列の左右から交互に 1 文字を取り出して結合する.	12 * 34 = 1324

A　WA の計算方法

4 章で述べたように，学習曲線を定量的に評価するために，加重平均精度 (WA) を導入した．この指標は，初期の学習ステップにおける正解率に高い重みを与えることで，汎化の効率（容易さ）を定量的に評価するものである．具体的には，以下の式で重み w_i（i はモデルの更新回数）を算出した．

$$w_i = -ai + w_{\max} \tag{1}$$

$$a = \frac{(w_{\max} - w_{\min})}{N} \tag{2}$$

$$w_{\min} = \frac{2}{(N+1)(\alpha+1)} \tag{3}$$

$$w_{\max} = \alpha w_{\min} \quad , \tag{4}$$

ここで，N はモデルの更新回数，α は初期の正解率にどれだけ重み付けするかを決めるハイパーパラメータである．この時，$\sum_{i=0}^{N} w_i = 1$ となる．本論文における，全ての実験では $\alpha = 1000$, $N = 100$ とした．

B　学習設定

事前学習済みモデルとして T5 [10] (v1.1) を使用した [1)]．モデルのパラメータ数は base，large，x-large それぞれ，2.5 億，8 億，30 億である．

T5 のファインチューニングの設定に従い，最適化器には Adafactor [16] を使用し，学習率のスケジューラは利用していない．学習率は事前学習時は 1.0×10^{-5}，ファインチューニング時はで 5.0×10^{-5}，バッチサイズは 32 とした．各モデルの学習は，NVIDIA A6000(48GB)，A100(80GB) で行った．

事前学習:　事前学習を行うため，各ドメイン (基本演算) ごとに 100,000 事例のデータを用意し，それらを結合したデータによってモデルの学習を行った．ただし，生産性に対する汎化能力の評価の際には評価ドメインの数式よりも短い (構成要素が 1 つ少ない) データも含めて事前学習を行った．事前学習は 検証データセットにおける正解率が 100%となるか，5 epoch の間正解率の改善が見られなくなるまで続けた．

事前学習の結果，全ての設定において事前学習に用いたデータセットと同じドメインの評価データセット (2,000 事例) において，少なくとも 99.5%の正解率に達した．

ファインチューニング:　ファインチューニング時には，各ドメインごとに 3,200 事例のデータを用意した．これは WA を計算するために，モデルを 100 回更新する時に要するデータ数である．(バッチサイズ 32 × モデルのアップデート回数 100)

C　データセットの詳細

各設定において，数式に登場する数値は 0 から 99 までである．また，式の順番は任意である（必ずしも最初の式が先に計算されるべきとは限らない）ことに注意されたい．算術演算の演算子としては, 算術の加算+, 減算-, 右の数のうち大きい方を返す max, 左右の数のうち小さい方を返す min の 4 種類としている．また，文字列演算の詳細については表 3 に示す．

1)　https://huggingface.co/docs/transformers/model_doc/t5v1.1