ニューラルネットが学習する意味表現は体系性を持つか

谷中 瞳 [1] 峯島 宏次 [2] 乾 健太郎 [3,1]
[1] 理化学研究所 [2] 慶應義塾大学 [3] 東北大学
hitomi.yanaka@riken.jp minesima@abelard.flet.keio.ac.jp inui@ecei.tohoku.ac.jp

1 はじめに

大規模計算環境の発展に伴い，大量のテキストデータを扱うことが可能となり，機械翻訳や質問応答といった様々な自然言語処理タスクにおいて，深層ニューラルネット (DNN) を用いた手法 [1, 2] が高精度を達成しつつある．一方で，DNN がデータセット中の表層的な関係やバイアスを学習しているという問題が指摘されており [3, 4]，DNN が自然言語の意味を学習できているのかは自明ではない．

DNN による表現学習の問題として，DNN による意味表現が，形式意味論で考えられてきた構成性 (compositionality) の原理—文全体の意味が構成要素の意味から統語構造に従って決まるという性質—を満たしているか，という問題がある．この問題は人間の認知のモデリングに関する 2 つの理論，すなわち，コネクショニズムと古典的計算主義の対立として古くから議論されている．その中で Fodor [5] は，*Ann loves Bob* という文の内容を思考できれば，*Bob loves Ann* という文の内容も思考できるというように，人間の認知能力には，ある処理ができれば関連した処理もできるという体系性 (systematicity) があることを主張している．Yanaka ら [6] は，この体系性の問題を再考し，現行の DNN が自然言語推論 (Natural Language Inference, NLI) を体系的には学習していないことを実証的に示した．しかし，DNN がなぜデータから NLI を体系的に学習できないのか，具体的には，DNN が文の意味を体系的に学習できないのか，それとも文間の推論に必要な規則を学習できないのかについては，特定できていなかった．

そこで本論文では，文から意味表示に変換する意味解析のタスクを用いて，DNN が文の構成的な意味を体系的に学習できているのか分析を行う．本論文の貢献は，(i) 意味解析で DNN の体系性を評価する方法の提案，(ii) 現行の DNN の意味における汎化性能の範囲の特定，の二点である．評価に用いるコードは研究利用が可能な形式で公開予定である．

2 関連研究

近年，SCAN [7] や CLUTRR [8] といった，自然言語の命令を入力とし命令に対応する行動を示す記号系列を出力する構成的なタスクで，DNN の汎化性能を評価するデータセットが数多く提案されている．しかし，これらは単純な人工タスクを扱っており，より複雑な構成性をもつ自然言語の意味における DNN の汎化性能については自明ではない．そこで Yanaka ら [6] は重要な推論現象の一つである monotonicity [9] に着目して，DNN の NLI における体系性を分析する手法を提案し，現行の DNN が NLI を体系的には学習していないことを示したが，体系的に学習できない原因は特定できていなかった．

また，DNN の自然言語における文法性を分析する先行研究として，名詞と動詞の数の一致 (subject-verb agreement) に着目した分析 [10] や，自然言語文から述語項構造への変換タスクを用いた分析 [11] がある．しかし，これらの先行研究では数の一致や受動態の文を学習し能動態の文に汎化するかといった形態論的・統語論的な汎化を中心に分析しており，否定や量化といった意味論的な汎化は扱っていない．また，[11] では 1 種類の意味表示を扱っており，評価指標も正解との完全一致率に限定されていた．意味表示は $A \wedge B$ と $B \wedge A$ が同じ意味を表すように，形式が違っていても同じことを表現しうるため，意味表示の機能に基づいて評価する必要がある．そこで本研究では意味論的な汎化を分析対象として，複数の意味表示の形式と複数の評価指標を用いることで，意味解析における DNN の総合的な分析を行う．

3 分析手法

3.1 概要

本研究では先行研究 [6] の (i) 未知の組合せと (ii) 未知の深さという 2 つの体系性の評価の観点を用いて，DNN が文から意味表示への変換を体系的に学

表 1　意味解析データセットの例．学習 1・2 は学習データ 1・2 の略．中央・非中央は埋め込み節の種類．

学習				テスト
言語現象		入力文	正解の意味表示	入力文
3.2. 未知の組合せにおける体系性（意味表示の例は FOL）				
基本量化表現 + 項		*One* tiger ran	$\exists x_1.(\mathbf{tiger}(x_1) \wedge \mathbf{run}(x_1))$	存在 + 形容詞
学習 1	形容詞	*One* **small** tiger ran	$\exists x_1.(\mathbf{small}(x_1) \wedge \mathbf{tiger}(x_1) \wedge \mathbf{run}(x_1))$	*A* **small** tiger ran
	副詞	*One* tiger ran **quickly**	$\exists x_1.(\mathbf{tiger}(x_1) \wedge \mathbf{run}(x_1) \wedge \mathbf{quickly}(x_1))$	数量 + 副詞
	結合子	*One* tiger ran **or came**	$\exists x_1.(\mathbf{tiger}(x_1) \wedge (\mathbf{run}(x_1) \vee \mathbf{come}(x_1)))$	*Two* tigers ran **quickly**
学習 2	存在	*A* tiger ran	$\exists x_1.(\mathbf{tiger}(x_1) \wedge \mathbf{run}(x_1))$	全称 + 結合子
	数量	*Two* tigers ran	$\exists x_1.(\mathbf{two}(x_1) \wedge \mathbf{tiger}(x_1) \wedge \mathbf{run}(x_1))$	*Every* tiger ran **or came**
	全称	*Every* tiger ran	$\forall x_1.(\mathbf{tiger}(x_1) \rightarrow \mathbf{run}(x_1))$	
3.3. 未知の深さにおける体系性（意味表示の例は VF）				
学習 1	深さ 0	Two dogs loved Ann	TWO DOG EXIST ANN LOVE	深さ 2：中央 + 非中央
学習 2	深さ 1：中央	Two dogs [that all cats kicked] loved Ann	TWO AND DOG ALL CAT INV KICK EXIST ANN LOVE	Two dogs [that a bear [that chased all polite cats] kicked] loved Ann
	深さ 1：非中央	Bob liked a bear [that chased all polite cats]	EXIST BOB EXIST AND BEAR ALL AND CAT POLITE CHASE LIKE	

習できているのかについて分析を行う．言語現象の網羅性と文の自然さをコントロールするため，分析には文脈自由文法 (CFG) の生成規則によって自動構築したデータセットを用いる．まず，CFG の生成規則（付録の表 7 参照）に従って，文と CFG 構文木を生成する．次に，CFG 構文木と単語への意味割り当てに基づいて，ラムダ計算によって意味表示を合成する．提案手法によって自動構築したデータの例を表 1 に示す．

また，文を入力として文を出力する Autoencoder の設定をベースラインとして意味解析の予測精度と比較し，文のエンコードの汎化に限界があるのか，意味表示へのデコードの汎化に限界があるのかを考察する．意味表示には一階述語論理に基づく形式（以下，FOL）と，括弧と変数を除去した形式 [12, 13] (Variable-Free form，VF) の 2 種類の形式を用いて，意味表示の形式の違いと学習の難しさの関係を分析する．例えば，(1) の文には次の 2 種類の意味表示（FOL, VF）が割り当てられる．

(1)　All white dogs ran
FOL: $\forall x_1.(\mathbf{white}(x_1) \wedge \mathbf{dog}(x_1) \rightarrow \mathbf{run}(x_1))$
VF: ALL AND WHITE DOG RUN

3.2　未知の組合せにおける体系性

本節では，DNN が量化表現と修飾表現の組合せからなる文の意味を体系的に学習できているかについて分析する手法を紹介する．この分析では，量化表現と修飾表現の組合せの学習に最低限必要なデータを学習データとしてモデルを訓練し，学習データにない未知の組合せをテストデータとしてモデルを評価する．

表 1 の例を考えよう．ここでは存在量化の一つ *one* を**基本量化表現**，*tiger* を**基本の項**として固定し，これに形容詞，副詞，結合子の 3 タイプ×各 5 語の合計 15 語の修飾表現を組み合わせた文の集合を**学習データ 1** とする．また，基本量化表現を含め存在量化 (*a*, *one*)，数量表現 (*two*, *three*)，全称量化 (*every*, *all*) の 3 タイプ×各 2 語の合計 6 語の量化表現と，基本の項との組合せから構成される文の集合を**学習データ 2** とする．モデルが学習データ 1 と学習データ 2 からの学習で体系性を獲得していれば，テストデータ中のあらゆる量化表現と修飾表現の組合せからなる文の意味を正しく予測できるはずである．本論文では，組合せの訓練に十分なデータサイズとして合計 50,000 件のデータを用意し，その中で基本量化表現に応じて学習データ約 12,000 件とテストデータ約 38,000 件に分割する．

3.3　未知の深さにおける体系性

体系性の性質の一つに，有限の文法規則から無限の文を生成できるという生産性 (productivity) がある．DNN が生産性を捉えていれば，文の構造がどれ

表 2　修飾表現のタイプ別の完全一致率の評価結果.

種類	GRU			Transformer		
	文	FOL	VF	文	FOL	VF
形容詞	47.5	18.9	42.3	5.0	26.8	27.6
形容詞＋否定	43.3	18.8	39.7	8.0	23.1	27.5
副詞	57.9	20.1	58.4	32.2	36.2	50.7
副詞＋否定	63.3	26.9	67.2	47.0	50.7	62.1
結合子	67.2	28.9	72.9	52.3	54.3	65.9
結合子＋否定	70.7	33.6	74.9	57.3	60.1	69.1

だけ深くなっても，構造の規則性から文の意味を計算できるはずである．そこで本節では，モデルが生産性を学習できているか分析する手法を紹介する．

この分析では，表 1 の例のように埋め込み節を含まない文を深さ 0，埋め込み節を 1 つ含む文を深さ 1 と呼び，深さ 0 と深さ 1 のデータを学習データとしてモデルを訓練する．そして，学習データよりも深い埋め込み節を含むテストデータでモデルを評価する．本論文ではテストデータの深さは最大 4 とし，学習に十分なデータサイズとして各深さ 20,000 件のデータを用意する．また，深さ以外の語彙と構文規則は学習データとテストデータで共通にする．

4　実験

4.1　実験設定

自然言語における汎化性能の評価に用いられる標準的なモデルとして，GRU に基づく seq2seq [14] と，Transformer に基づく seq2seq [15] の 2 種類のモデルを評価した．実装には PyTorch を用いた．GRU のエンコーダ，デコーダは 1 層の単方向 GRU[1] を用い，隠れ状態は 256 次元とした．Transformer のエンコーダ，デコーダは共に 3 層とし，モデルサイズは 512 次元，隠れ状態は 256 次元とした．上記以外のパラメータはすべて共通で，埋め込み層は 256 次元，ドロップアウトの確率は 0.1，ミニバッチサイズは 128 とし，最適化手法には初期学習率を 0.0005 とした Adam [16] を使用した．各実験を 5 回ずつ行い，平均の精度を最終的なモデルの評価結果とする．

4.2　評価指標

モデルが予測した FOL の意味表示は，(i) 正解の意味表示との完全一致率に加えて，意味表示の構造や機能を考慮した評価方法として，(ii) 自動定理証明 (automated theorem proving, ATP) による評価，(iii) monotonicity に基づく評価の計 3 種類の方法で評価する．

ATP による評価　一階述語論理の定理証明器 vampire[2] を用いて，(i) 正解の意味表示 G が予測の意味表示 P を含意するか ($G \Rightarrow P$)，(ii) 予測の意味表示 P が正解の意味表示 G を含意するか ($G \Leftarrow P$)，という双方向の含意関係の証明を試み，評価を行う．

monotonicity に基づく評価　予測した意味表示が否定や量化のスコープを正しく捉えられているか分析するため，論理式中の各述語に対して極性 (polarity) を計算し，適合率，再現率，F 値を評価する．極性は FOL の意味表示から計算でき，(2) のように存在量化 ($\exists$) の中では upward monotone ($\uparrow$)，(3) のように条件 ($\rightarrow$) の前件や否定 ($\neg$) のスコープの中では downward monotone ($\downarrow$) と計算できる．また，(4) の *dogs* のように downward monotone に 2 回埋め込まれると，さらに極性が反転する．

(2)　One dog ran: $\exists x.(\mathbf{dog}^{\uparrow}(x) \wedge \mathbf{run}^{\uparrow}(x))$

(3)　All dogs ran: $\forall x.(\mathbf{dog}^{\downarrow}(x) \rightarrow \mathbf{run}^{\uparrow}(x))$

(4)　All dogs didn't run: $\neg\forall x.(\mathbf{dog}^{\uparrow}(x) \rightarrow \mathbf{run}^{\downarrow}(x))$

表 3　量化表現のタイプ別の完全一致率の評価結果.

種類	GRU			Transformer		
	文	FOL	VF	文	FOL	VF
存在	94.5	96.1	99.7	16.3	99.9	100.0
数量	27.4	7.6	37.0	20.8	18.1	20.7
全称	59.8	3.1	39.5	19.1	8.3	17.7
valid	99.9	98.2	99.6	100.0	100.0	100.0

4.3　実験結果

未知の組合せにおける体系性　表 2 に修飾表現のタイプ別の完全一致率による評価結果を示す．両モデルとも結合子 > 副詞 > 形容詞の順に精度が高く，否定の有無については特別な傾向はない．これは表 1 の例のように，形容詞を追加すると後続の語の位置がずれるのに対して，副詞や結合子を文末に追加した場合は語の位置が変化せず，文の構造を記憶しやすいことが影響していると考えられる．

表 3 に量化表現のタイプ別の完全一致率による評価結果を示す．GRU と Transformer の結果を見ると，基本量化表現である *one* と同じ存在量化の量化表現

1)　双方向よりも単方向のエンコーダの方が文の構造に頑健であるという報告 [11] があり，本論文では単方向を採用した．

2)　https://github.com/vprover/vampire

表 4　量化表現のタイプ別の ATP による FOL の評価結果. $G \Leftrightarrow P$ は双方向含意関係の証明の精度.

種類	GRU				Transformer			
	一致率	含意関係			一致率	含意関係		
		$G \Rightarrow P$	$G \Leftarrow P$	$G \Leftrightarrow P$		$G \Rightarrow P$	$G \Leftarrow P$	$G \Leftrightarrow P$
存在	96.1	99.8	100.0	99.8	99.9	100.0	100.0	100.0
数量	7.6	77.1	19.0	10.4	18.1	91.3	21.1	12.4
全称	3.1	7.1	18.7	2.7	8.3	21.1	83.4	12.3

表 5　量化表現のタイプ別の monotonicity に基づく FOL の評価結果. prec：適合率，rec：再現率，f1：F 値.

種類	GRU						Transformer					
	upward			downward			upward			downward		
	prec	rec	f1	prec	rec	f1	prec	rec	f1	prec	rec	f1
存在	100.0	99.9	99.9	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0
数量	99.2	75.5	84.8	99.6	94.7	96.8	100.0	79.5	88.1	100.0	95.6	97.5
全称	92.6	89.9	90.9	42.9	39.4	40.7	97.3	93.4	94.9	79.4	70.0	73.4

表 6　未知の深さにおける評価結果.

深さ	GRU			Transformer		
	文	FOL	VF	文	FOL	VF
0, 1	100.0	100.0	100.0	100.0	100.0	100.0
2	15.8	0.31	0.32	0.23	0.60	0.58
3	0.21	0.02	0.04	0.03	0.12	0.12
4	0.03	0.00	0.00	0.00	0.02	0.02

に対してほぼ 100%の完全一致率であり，学習データ 1 の基本量化表現と同じタイプの量化表現を含む組合せには汎化しやすいことが示唆される．（他のタイプの量化表現を基本量化表現とした場合の結果は付録の表 8，表 9 参照.）意味表示間の違いとしては，FOL よりも括弧・変数がなく単純な形式である VF の方が汎化しやすい傾向がある．モデル間の違いとしては，Transformer は文を入力とし文を出力する Autoencoder の設定で精度が低く，文のエンコード時に汎化できていないと考えられる．

ATP による評価結果（表 4）と monotonicity に基づく評価結果（表 5）を見ると，完全一致率よりも精度が高い．これは括弧の数や等価な論理式の出現順序が正解とは異なるため，完全一致率では誤答とみなされるケースがあるからである．monotonicity で評価した場合は全称量化の downward の精度が低く，全称量化のスコープを学習することが難しいことを示している．さらに，ATP の全称量化の精度を見ると，GRU は双方向で低いのに対し Transformer では $G \Leftarrow P$ よりも $G \Rightarrow P$ の精度が低い．実際のエラーを見ると次のように GRU は修飾表現が丸ごと抜けているエラーが多く，Transformer は前件の位置が誤っているエラーが多いという違いが見られた．

入力文　Every wild cat escaped or ran

正解　$\forall x.((\mathbf{cat}(x) \wedge \mathbf{wild}(x)) \rightarrow (\mathbf{escape}(x) \wedge \mathbf{run}(x)))$

GRU　$\forall x.(\mathbf{cat}(x) \rightarrow (\mathbf{escape}(x) \wedge \mathbf{run}(x)))$

Trans　$\forall x.(\mathbf{cat}(x) \rightarrow \mathbf{wild}(x) \wedge (\mathbf{escape}(x) \wedge \mathbf{run}(x)))$

未知の深さにおける体系性　表 6 を見ると，両モデルとも未知の深さに対してはほぼ汎化できていない．また，文を入力とし文を予測する Autoencoder の設定では，GRU は深さ 2 を含む文を 15.8%予測できているが，両モデルとも精度が低く，文のエンコードが汎化できていないことが示唆される．

5　おわりに

本論文では，DNN がデータからの学習で文の構成的な意味における体系性を獲得しているかについて分析する手法を紹介した．実験の結果，GRU，Transformer ともに量化表現と修飾表現の組合せにおいて学習データと文の構造が変わらない場合は汎化しやすい一方で，未知の深さなど構造が変わる場合は文のエンコード時に汎化できていないことが示唆された．また，ATP や monotonicity で評価することでモデルのエラー傾向を特定でき，表面的な形式だけでなく意味表示の機能を考慮した指標で評価する必要性が示唆された．今後，SCAN などの人工タスクで高精度を達成しているモデル [17] を提案手法で分析し，構成性を満たす表現学習の検討を進める．

謝辞．本研究は理研・産総研「チャレンジ研究」(FS 研究)，JSPS 科研費 JP20K19868 の助成を受けたものである．

参考文献

[1] Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2019.

[2] Jacob Devlin, Chang Ming-Wei, Lee Kenton, and Toutanova Kristina. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, pp. 4171–4186, 2019.

[3] R. Thomas McCoy, Ellie Pavlick, and Tal Linzen. Right for the wrong reasons: Diagnosing syntactic heuristics in natural language inference. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 3428–3448, 2019.

[4] Nelson F. Liu, Roy Schwartz, and Noah A. Smith. Inoculation by fine-tuning: A method for analyzing challenge datasets. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, pp. 2171–2179, 2019.

[5] Jerry A. Fodor and Zenon W. Pylyshyn. Connectionism and cognitive architecture: A critical analysis. *Cognition*, Vol. 28, No. 1-2, pp. 3–71, 1988.

[6] Hitomi Yanaka, Koji Mineshima, Daisuke Bekki, and Kentaro Inui. Do neural models learn systematicity of monotonicity inference in natural language? In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 6105–6117, 2020.

[7] Brenden M. Lake and Marco Baroni. Generalization without systematicity: On the compositional skills of sequence-to-sequence recurrent networks. In *Proceedings of the 35th International Conference on Machine Learning (ICML)*, 2017.

[8] Koustuv Sinha, Shagun Sodhani, Jin Dong, Joelle Pineau, and William L. Hamilton. CLUTRR: A diagnostic benchmark for inductive reasoning from text. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 4506–4515, 2019.

[9] Johan van Benthem. Determiners and logic. *Linguistics and Philosophy*, Vol. 6, No. 4, pp. 447–478, 1983.

[10] Tal Linzen, Emmanuel Dupoux, and Yoav Goldberg. Assessing the ability of LSTMs to learn syntax-sensitive dependencies. *Transactions of the Association for Computational Linguistics*, Vol. 4, pp. 521–535, 2016.

[11] Najoung Kim and Tal Linzen. COGS: A compositional generalization challenge based on semantic interpretation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 9087–9105, 2020.

[12] Franz Baader, Diego Calvanese, Deborah McGuinness, Peter Patel-Schneider, Daniele Nardi, et al. *The Description Logic Handbook: Theory, Implementation and Applications*. Cambridge university press, 2003.

[13] Ian Prat-Hartmann and Lawrence S. Moss. Logics for the relational syllogistic. *The Review of Symbolic Logic*, Vol. 2, No. 4, pp. 647—-683, 2009.

[14] Kyunghyun Cho, Bart van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using RNN encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1724–1734, 2014.

[15] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems (NeurIPS)*, pp. 5998–6008, 2017.

[16] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2015.

[17] Yewen Pu, Kevin Ellis, Marta Kryven, Josh Tenenbaum, and Armando Solar-Lezama. Program synthesis with pragmatic communication. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems (NeurIPS)*, 2020.

A 付録

未知の組合せにおける体系性の評価の補足 *one* を基本量化表現，*dog* を基本の項としたときの学習データ 1 に含まれる文の例を (5) に示す．学習データ 1 を通して，モデルに修飾表現を含む文のパターンを一通り訓練させる．また，学習データ 2 に含まれる文の例を (6) に示す．学習データ 2 を通して，モデルに量化表現を含む文のパターンを一通り訓練させる．モデルが学習データ 1 と学習データ 2 を体系的に学習できていれば，(7) にあるような様々な量化表現と修飾表現の組み合わせからなる文についても正しい意味表示を予測できるはずである．

(5) a. **One** *small* dog ran
b. **One** dog ran *quickly*
c. **One** dog ran *or came*

(6) a. **One** dog ran
b. **A** dog ran
c. **Every** dog ran
d. **Two** dogs ran

(7) a. **A** *small* dog ran
b. **Every** dog ran *quickly*
c. **Two** dogs ran *or came*

表 7 にデータ構築に用いた生成規則と語彙項目を示す．語彙は分析のしやすさを考慮して自然かつ基本的な文を生成するよう一般名詞・固有名詞・自動詞・他動詞は各 10 語，量化表現は 6 語（存在量化，数量表現，全称量化それぞれ 2 語），形容詞・副詞は各 5 語を選定し，マルチワードは使用しない．

表 8 に数量表現の *two* を基本量化表現とした場合，表 9 に全称量化の *every* を基本量化表現とした場合の，量化表現のタイプ別の完全一致率の評価結果を示す．いずれも基本量化表現と同じタイプの量化表現を含む組合せの予測精度が他のタイプの予測精度よりも高くなっており，基本量化表現と同じタイプの量化表現を含む組合せには汎化しやすいことが示唆される．

未知の深さにおける体系性の評価の補足 基本量化表現 *one* を含む深さ 2 以上のケース 1000 件を学習データに追加した場合の結果を表 10 に示す．表 10 を見ると，表 6 と同様の結果であり，深さ 2 以上のケースを一部モデルに教えても深さ 2 以上は汎化できていないことがわかる．

表 7 データ構築に用いた生成規則と語彙項目（抜粋）．

文の生成規則		
S	→	NP VP | NP *did not* VP
VP	→	IV | IV Adv | IV or IV′ | IV and IV′ | TV NP
NP	→	PN | Q N | Q Adj N | Q N $\overline{\text{S}}$
$\overline{\text{S}}$	→	*that* TV NP | *that* NP TV | NP TV
語彙項目		
Q	→	{*every, all, a, one, two, three*}
N	→	{*dog, rabbit, cat, bear, tiger*}
PN	→	{*ann, bob, fred, chris, eliott*}
IV	→	{*ran, walked, swam, danced, dawdled*}
IV′	→	{*laughed, groaned, roared, screamed*}
TV	→	{*kissed, kicked, cleaned, touched*}
Adj	→	{*small, large, crazy, polite, wild*}
Adv	→	{*slowly, quickly, seriously, suddenly*}

表 8 量化表現のタイプ別の完全一致率の評価結果（数量表現の *two* を基本量化表現とした場合）．

種類	GRU			Transformer		
	文	FOL	VF	文	FOL	VF
存在	40.8	11.6	45.3	42.2	34.0	10.5
数量	83.0	59.5	42.8	12.3	99.9	80.9
全称	58.4	2.5	39.2	30.9	0.0	90.9
valid	100.0	84.3	98.9	100.0	100.0	100.0

表 9 量化表現のタイプ別の完全一致率の評価結果（全称量化の *every* を基本量化表現とした場合）．

種類	GRU			Transformer		
	文	FOL	VF	文	FOL	VF
存在	57.3	1.6	61.3	32.4	2.1	20.8
数量	28.8	1.4	69.3	67.1	0.1	99.7
全称	31.1	33.8	100.0	63.1	100.0	99.9
valid	98.3	93.4	100.0	100.0	100.0	99.9

表 10 未知の深さにおける評価結果（基本量化表現 *one* を含む深さ 2 以上のケースを学習データに加えた場合）．

深さ	GRU			Transformer		
	文	FOL	VF	文	FOL	VF
0, 1	100.0	100.0	100.0	100.0	100.0	100.0
2	10.33	0.98	1.33	6.03	4.89	8.73
3	0.42	0.15	0.25	0.29	0.42	0.48
4	0.10	0.00	0.00	0.15	0.15	0.15