メッセージの階層構造を把握するための parsing action がランダムではないのはなぜか？

加藤大地　上田亮　宮尾祐介
東京大学
{daichi5967, ryoryoueda, yusuke}@is.s.u-tokyo.ac.jp

概要

仮に人間がランダムに parsing action をとる言語理解をしていたならば、どんな階層構造を取っても理解できるような頑健な記号体系が作られていた可能性もあるが、人間の parse 戦略がそうなっていないのはなぜだろうか？実際に、階層的なバイアスを持つモデルによる言語創発の先行研究では、メッセージをランダムに parse する、人間の言語理解と乖離した戦略をとるエージェントのコミュニケーション精度が高くなることが報告された。本研究では、(I) ランダムな parse 戦略では意味の理解が難しくなるような、階層的構造を持つより複雑な入力を用いる、(II) 言語の単語や文字の順序に影響を与えるとされる surprisal に関する項を目的関数に組み込む、というシンプルで自然な変更を実験設定に加え、ランダムな parse 戦略を取るエージェントのコミュニケーション精度がどうなるか、検証を行った。

1　はじめに

言語創発 (Emergent Communication; EC) [1] は、構成論的アプローチを用いて人間の言語の起源を探る、計算言語学の一分野である。ニューラルネットワークで構成されるエージェント同士に、協調的なタスクを解かせる中で記号列を介した会話をさせ、学習の中で生じるその記号列の集合を、ある種の言語—**創発言語**とみなし、人間の言語と比較・解析するのが、本分野の主な目的の 1 つである。

EC においてエージェント間に言語を創発させるには、シグナリングゲーム [2] と呼ばれる枠組みを使うことが多い。シグナリングゲームでは、**sender** と **receiver** という 2 種類のエージェントが登場する。環境として、**意味空間** $\mathcal{X}$、**メッセージ空間** $\mathcal{M}$ を準備する。まず、ランダムにサンプルされた入力 $x \in \mathcal{X}$ が sender のみに渡され、sender は x をもとにメッセージ m を生成する。その後 m が receiver に渡され、receiver は m のみをもとに sender が受け取っていた x を予測して、予測 $\hat{x}$ を生成する。この意味で、sender はパラメタ ϕ を用いて $S_\phi(M \mid X)$、receiver はパラメタ θ を用いて $R_\theta(X \mid M)$ とモデル化される。最後に入力 x と予測 $\hat{x}$ を比較し、一致していれば正の報酬が両エージェントに与えられる。この一連のプロセスが、シグナリングゲームの 1 イテレーションを構成する。このプロセスを繰り返し行い強化学習することで、双方のエージェントにとって意味のある創発言語が生じるようになる。

EC では、LSTM [3]、GRU [4]、Transformer [5] などの sequential なモデルがほとんどの研究で用いられる。sequential なモデルは、NLP やその他の応用分野において多くの実用的な成功を収めてはいるが、人間の言語学習に比べ極端にデータ効率が悪い等 [6] の指摘もあり、人間をモデル化するものとして適切かは疑問が残る。これに対し、人間の言語に木構造などの階層的な構造を仮定して議論する研究は古くから盛んに行われており [7]、EC 分野外においては、人間の言語処理の認知プロセスを階層的に扱うことの妥当性は考察され続けている [8]。

このような背景から、加藤ら [9] は、階層的なバイアスを明示的に持つアーキテクチャを、receiver のメッセージを受け取る部分 **(メッセージエンコーダ)** に組み込み、EC にどのような影響があるか検証する実験を行った。彼らは、未知のデータを用いた際のコミュニケーションの成功率 **(Comunication Accuracy; ComAcc)** を調べ、階層的なモデルの ComAcc が他モデルに比べて高くなるような条件について議論した。その中で彼らは、**random-branching** と呼ばれるベースラインを独自に定義した。これはメッセージの階層構造を把握するための parsing action をランダムに予測する、人間の言語理解とは乖離した戦略を持つにも関わらず、

ComAcc が非常に高くなるという結果を得た。

この結果は一見直感に反するように思えるが、コミュニケーションの精度を上げるということだけに焦点を当てた場合には、メッセージの階層構造をどう捉えるかに意味が寄らない記号体系は頑健であるという点において、むしろ良い選択肢であると言えるかもしれない。しかし、現実の人間の parsing action を決める戦略は、そうはなっていない。この原因を探る一助になり得る考察として、加藤ら [9] は、実験で用いた意味空間が、ランダムにメッセージの階層構造を予測しても意味が伝達してしまうほど簡潔なものになっており、意味空間が十分に複雑でないことがこの結果の要因の 1 つである可能性を指摘した。また、人間の言語の単語・文字順序に影響を与えるとされる surprisal [10] の側面がモデル化されていないことも、要因として考えられるとした。

この見解を踏まえ本研究では、**Stack LSTM** [11, 12] というモデルを使って、以下の 2 つの自然で最小限な条件を加えた実験を行い、random-branching の ComAcc が下がるか否か、下がらなかったとしても、人間の言語理解をモデル化するものとして不適切な結果が生じるか、ということを検証する[1)]。

実験 I：より階層的な構造を持った意味空間を用いる. 加藤ら [9] が用いた attribute-value 形式の代わりに、より階層的な構造を持つ Dyck-k という文脈自由言語を意味空間として用いた実験を行う。

実験 II：surprisal の側面を組み込んだ枠組みを用いる. シグナリングゲームを (beta-)VAE [14, 15] として再解釈すると、surprisal の側面を自然に組み込むことができるという理論 [13] を用いた実験を行う。

実験 I では、attribute-value 形式より複雑な意味空間を入力として用いることで、random-branching の ComAcc が低くなる傾向が見られ、人間のような parsing action の戦略が生まれるためには複雑な意味空間が重要であるという示唆を得られる結果となった。実験 II では、surprisal の側面をモデル化しても ComAcc を比較した際の傾向は変わらないことが分かった。しかし、random-branching が未知の意味を理解しようとする際に、人間の言語理解では通常忌避される高い認知負荷を受けていることも分かり、人間をモデル化するエージェントとして不適であることを説明する一助となる結果も得た。

1) 実験 II で用いる理論 [13] の都合上、先行研究で用いられたモデルは確率モデルが異なり使えないため、別のモデルとして Stack LSTM を使うこととなった。Stack LSTM は予備実験 (付録 B を参照) を通じて、概ね先行研究と同様の結果が得られることが分かっている。実験 I に関しては、RL-SPINN でも実験することができるが、今回は簡単のため、Stack LSTM を統一的に使って実験を行うことに注意されたい。

2 背景

2.1 Stack LSTM

Stack LSTM は、通常の LSTM を Neural Stack [11] と呼ばれる記憶構造で拡張したモデルである。従来の stack は、pop と push という 2 つの離散的なアクションを持ち、そのアクションに従って離散的に内部状態を更新する。そのため、学習時に通常の backpropagation が使えないことから、ニューラルネットワークのアーキテクチャの一部として stack を使うことは一般に難しかった。

Grefenstette ら [11] は、pop と push を連続値として扱い、中間状態の stack を作ることで、stack 全体の演算を微分可能にした。Neural Stack は、ある時間 t において、ベクトル列 $\boldsymbol{V}_t := (\boldsymbol{V}_t[1], \cdots, \boldsymbol{V}_t[t])$ とスカラー列 $\boldsymbol{s}_t := (\boldsymbol{s}_t[1], \cdots, \boldsymbol{s}_t[t])$ という状態を持つ。$\boldsymbol{V}_t[i]$ は stack に積まれている（下から数えて）i 番目の要素、$\boldsymbol{s}_t[i]$ は $\boldsymbol{V}_t[i]$ がどれほど stack に残っているか、その「強さ」を表す。

Neural Stack はある時間 t において、入力として、一つ前の stack の状態 $\boldsymbol{V}_{t-1}, \boldsymbol{s}_{t-1}$ と、stack に積まれるベクトル $\boldsymbol{v}_t$、pop、push、read の強さ u_t, d_t, r_t を受け取る。一ステップの中で Neural Stack は、pop、push、read という操作をこの順番で行う。pop は、stack の一番上の要素の強さを u_t 分だけ減らす操作である。一番上の要素の強さが足りなければその下の要素から、さらにそれでも足りなければその下の要素から、というように、再帰的にベクトルを取り除く。push は単純で、stack の一番上に要素 $\boldsymbol{v}_t$ を強さ d_t で置く操作である。read は pop と同様に、r_t 分に達するまで、上から順にベクトルを読んでいく操作である。それぞれの強さで重み付けして複数の要素の和をとり、その和が t での read ベクトル $\boldsymbol{r}_t$ となる。

さらに、Grefenstette ら [11] は、Neural Stack への入力を制御する RNN 型の controller という機構を加え、controller と Neural Stack を組み合わせた Stack RNN というモデルを提案した（図 1を参照）。特に、controller として LSTM を用いたものを、**Stack LSTM** と呼ぶことにする。なお、本実験では、pop、push、read の値として 1 以上の値も取れるようにした Merrill ら [12] の Stack LSTM の変種を用いる。

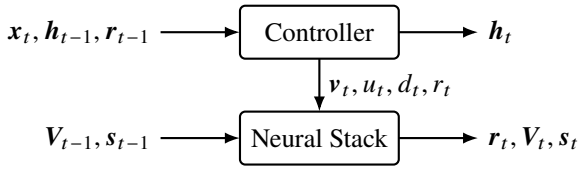


図 1: Stack RNN の概略図

2.2 シグナリングゲームを VAE として再解釈する枠組み

上田ら [13] は、シグナリングゲームにおける sender と receiver の役割が、VAE における encoder と decoder の役割に非常によく合致することに着目し、receiver を意味とメッセージの同時分布として再定義した：

$$R_\theta^{\text{joint}}(X, M) := P_\theta^{\text{prior}}(M) R_\theta(X \mid M)$$

ここで、$P_\theta^{\text{prior}}(M)$ はメッセージの事前分布である。さらに、beta-VAE [15] の目的関数を参考に、シグナリングゲームの目的関数 $\mathcal{J}_{\text{ec-vae}}$ を

$$\mathcal{J}_{\text{ec-vae}} = \mathbb{E}_{x \sim P_{\text{inp}}(X)} \Big[\mathbb{E}_{m \sim S_\phi(M \mid x)} [\log R_\theta(x \mid m)] - \beta D_{\text{KL}}(S_\phi(M \mid x) \,\|\, P_\theta^{\text{prior}}(M)) \Big]$$

として再定義した[2)]。彼らは、この目的関数を式変形することで、メッセージから意味を再構成する項と、surprisal が小さくなるようにする項とのトレードオフが、目的関数に自然に組み込めていることを示した。

3 実験手法

本研究の実験手法を簡潔に説明する。詳細な設定は、付録 A を参照すること。

3.1 実験 I

実験設定： 実験設定は基本的に先行研究 [9] に倣っており、変更した点は主に (1) メッセージエンコーダにおいて、RL-SPINN の代わりに、Stack LSTM [12] を用いる、(2) 意味空間として、Dyck-k を使う、という 2 点である。Dyck-k は、k 種類の括弧が正しい順番で閉じられているような文が集まった言語であり、厳密には以下の文脈自由文法で定義される：

$$S \to ({}_i\ S\ )_i\ S \mid \varepsilon \quad (1 \le i \le k)$$

なお、意味空間の大きさを限定するため、本実験では、文の長さを $l_{\max}$ で制限する。意味空間として試す設定は $(k, l_{\max}) = (1, 18), (4, 8), (9, 6)$ とする。

ベースライン： 本実験では、階層性のバイアスを明示的にもつ我々の Stack LSTM ベースのモデルに加え、ベースラインとして、left-branching と random-branching を用意する[3)]。公正な比較を行うため、モデル間の唯一の違いは、pop、push、read の強さの決め方である。我々のモデルでは、LSTM controller がそれぞれ $u_t \sim [0, k_u]$, $d_t \sim [0, k_d]$, $r_t \sim [0, k_r]$ の範囲で自由に強さの値を決める。これに対し、left-branching は、完全に左寄りの木が構成されるように、つまり従来の sequential なモデルと同じ順序でメッセージが処理されるように、$u_t = 1$, $d_t = 1$, $r_t = 1$ で固定する。また、random-branching は、LSTM controller と同じ範囲の連続一様分布からそれぞれランダムに強さをサンプルする。なお、同じメッセージが来たとしても、強さをランダムにサンプルすることに注意されたい。

3.2 実験 II

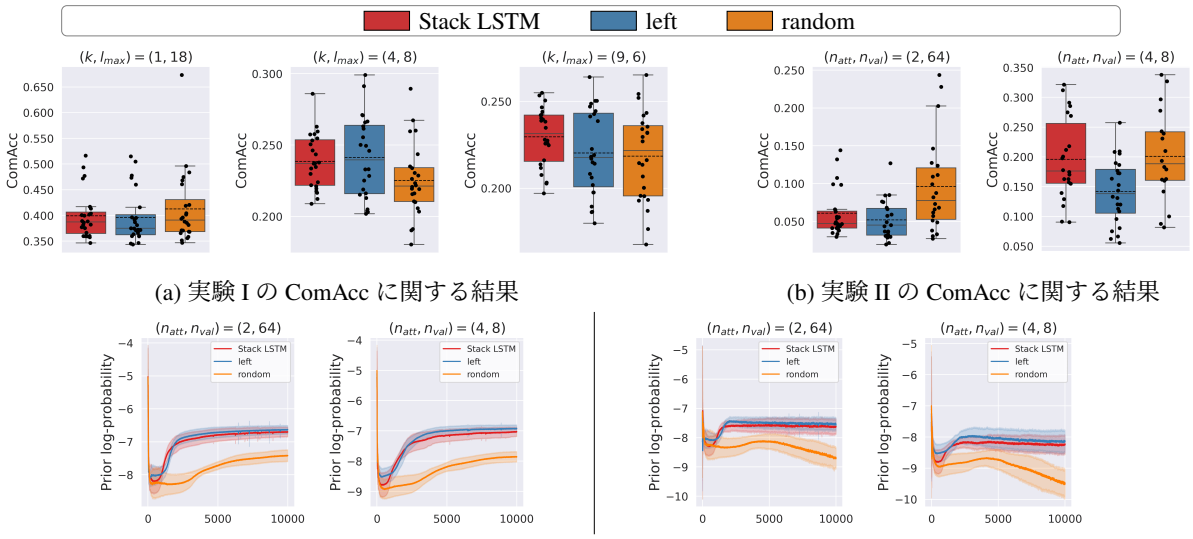
実験設定： 基本的な実験設定は実験 I に倣う。2.2 節で説明したように、receiver は、意味の再構成に加え、メッセージの事前分布の予測も行う必要がある。具体的には、各タイムステップ t において、その前のステップの read ベクトル $\boldsymbol{r}_{t-1}$ を用いて、$P_\theta^{\text{prior}}(M_t \mid M_{1:t-1})$ を予測するアーキテクチャを receiver に加える。最終のタイムステップに到達した後、そのゲームにおけるメッセージ全体の確率 $P_\theta^{\text{prior}}(M) = \prod_{t=1}^{|M|} P_\theta^{\text{prior}}(M_t \mid M_{1:t-1})$ を計算する。意味空間は先行研究 [9] に倣い、attribute-value 形式を用いる。試す設定は $(n_{\text{att}}, n_{\text{val}}) = (2, 64), (4, 8)$ とする。なお、n_{att} は attribute の数、n_{val} は各 attribute が取りうる値の種類の数を表す。

4 実験結果と考察

全ての実験において、それぞれの設定ごとに 24 の異なるランダムシード値を用いて実行を行った。その結果をまとめたものが 図 2 である。なお実験 II では、最終のイテレーションにおいて、KL ダイバージェンスにかかる係数 β の値が 0.95 以上でない実行を除外した結果を掲載している。

2) 上田ら [13] は、KL ダイバージェンスにかかる係数 β に関して、REWO [16] を用いて徐々に 1 に近づけるアニーリングを行った。本実験もこの設定に倣う。

3) ベースラインの名前は、先行研究 [9] に倣った。



(a) 実験 I の ComAcc に関する結果

(b) 実験 II の ComAcc に関する結果

(c) 実験 II の $\log P_\theta^{\text{prior}}(M)$ に関する結果。横軸はイテレーション数、縦軸はメッセージに対する確率の対数を表す。左 2 つの図は学習データに対する結果、右 2 つの図はテストデータに対する結果である。

図 2: 実験結果

4.1 実験 I

図 2a の通り、k の値が大きい設定では、random-branching の ComAcc が他のモデルに比べて低くなる傾向が見られる。Dyck-k において、$k = 1$ の設定では、各タイムステップにおいて開いている括弧の数、つまりネストの深さだけを覚えておけば良いため、意味空間として実質的に階層性を持たず、random-branching でもある程度簡単に扱えると考えられる。それに対し、k が 1 よりも大きい設定では、ネストの深さのみならず、どの括弧がどの順番で開いているかの情報を正確に記憶する必要があり、正確にメッセージの階層構造を保持することが難しい random-branching の ComAcc が低くなる傾向が出たと考えられる。この結果は、EC の実験において、attribute-value 形式よりもより複雑な意味空間を使うことが有用であることを示すものとなり得る。

4.2 実験 II

図 2b の通り、surprisal に関する項を入れること自体は、random-branching の ComAcc が高くなるという傾向に大きな違いをもたらすことは無かった。

しかし、図 2c の通り、我々の Stack LSTM ベースのモデルや sequential なモデルを模した left-branching では、学習時のメッセージの予測可能性が上がるにつれて、テスト時のメッセージの予測可能性も向上している。それに対し、random-branching では、学習時のメッセージの予測可能性が上がるにつれ、テスト時のメッセージの予測可能性が下がるという、ある種の過学習が発生している。surprisal の理論の観点から議論すると、未知の意味を理解しようとする際、random-branching を用いた receiver は、強い「驚き」とともにメッセージの意味を理解している。このような高い認知コストに常に晒されているモデルは人間的とは言えず、この観点は random-branching を使うことの不適性を説明する方法の 1 つとなり得る。

5 おわりに

本実験では、メッセージの階層構造を予測する際にランダム性を持つエージェントの ComAcc が不当に高くなるという加藤ら [9] の実験結果を踏まえ、彼らの見解を実験的に検証した。実験 I では、階層的な構造を持つ、より複雑な意味空間を用いることで、attribute-value 形式を用いたときよりも、ランダムなエージェントとそれ以外のモデルとの ComAcc の差が縮まる傾向を観察した。この結果から、EC において、階層的な構造を持つ意味空間を用いることの意義が示唆された。実験 II では、surprisal の側面を環境に組み込むことで、ランダムなエージェントが、未知の意味に対するメッセージを理解する際に、人間のコミュニケーションでは通常避けられる高い認知負荷を受けている状態を確認した。この結果は、ランダムなエージェントを人間のモデルとして用いることの不適切さを説明する事実の 1 つとなり得る。

謝辞

本研究は JSPS 科研費 JP23KJ0768、JST ACT-X（JPMJAX24C5）の助成を受けたものです。

参考文献

[1] Angeliki Lazaridou and Marco Baroni. Emergent multi-agent communication in the deep learning era. **CoRR**, Vol. abs/2006.02419, , 2020.

[2] David K. Lewis. **Convention: A Philosophical Study**. Wiley-Blackwell, 1969.

[3] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. **Neural Comput.**, Vol. 9, No. 8, pp. 1735–1780, 1997.

[4] Kyunghyun Cho, Bart van Merrienboer, Çaglar Gülçehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using RNN encoder-decoder for statistical machine translation. In Alessandro Moschitti, Bo Pang, and Walter Daelemans, editors, **Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL**, pp. 1724–1734. ACL, 2014.

[5] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett, editors, **Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA**, pp. 5998–6008, 2017.

[6] Alex Warstadt, Aaron Mueller, Leshem Choshen, Ethan Wilcox, Chengxu Zhuang, Juan Ciro, Rafael Mosquera, Bhargavi Paranjabe, Adina Williams, Tal Linzen, et al. Findings of the babylm challenge: Sample-efficient pretraining on developmentally plausible corpora. In **Proceedings of the BabyLM Challenge at the 27th Conference on Computational Natural Language Learning**, 2023.

[7] Noam Chomsky. **Syntactic Structures**. De Gruyter Mouton, Berlin, Boston, 1957.

[8] Matthew J Nelson, Imen El Karoui, Kristof Giber, Xiaofang Yang, Laurent Cohen, Hilda Koopman, Sydney S Cash, Lionel Naccache, John T Hale, Christophe Pallier, et al. Neurophysiological dynamics of phrase-structure building during sentence processing. **Proceedings of the National Academy of Sciences**, Vol. 114, No. 18, pp. E3669–E3678, 2017.

[9] Daichi Kato, Ryo Ueda, Jason Naradowsky, and Yusuke Miyao. Emergent communication with stack-based agents. In **Proceedings of the 46th Annual Meeting of the Cognitive Science Society**, 2024.

[10] Roger Levy. Expectation-based syntactic comprehension. **Cognition**, Vol. 106, No. 3, pp. 1126–1177, 2008.

[11] Edward Grefenstette, Karl Moritz Hermann, Mustafa Suleyman, and Phil Blunsom. Learning to transduce with unbounded memory. In Corinna Cortes, Neil D. Lawrence, Daniel D. Lee, Masashi Sugiyama, and Roman Garnett, editors, **Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada**, pp. 1828–1836, 2015.

[12] William Merrill, Lenny Khazan, Noah Amsel, Yiding Hao, Simon Mendelsohn, and Robert Frank. Finding syntactic representations in neural stacks. **CoRR**, Vol. abs/1906.01594, , 2019.

[13] Ryo Ueda and Tadahiro Taniguchi. Lewis's signaling game as beta-vae for natural word lengths and segments. In **The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024**. OpenReview.net, 2024.

[14] Diederik P. Kingma and Max Welling. Auto-encoding variational bayes. In Yoshua Bengio and Yann LeCun, editors, **2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings**, 2014.

[15] Irina Higgins, Loïc Matthey, Arka Pal, Christopher P. Burgess, Xavier Glorot, Matthew M. Botvinick, Shakir Mohamed, and Alexander Lerchner. beta-vae: Learning basic visual concepts with a constrained variational framework. In **5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings**. OpenReview.net, 2017.

[16] Alexej Klushyn, Nutan Chen, Richard Kurle, Botond Cseke, and Patrick van der Smagt. Learning hierarchical priors in vaes. In Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d'Alché-Buc, Emily B. Fox, and Roman Garnett, editors, **Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada**, pp. 2866–2875, 2019.

[17] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In Yoshua Bengio and Yann LeCun, editors, **3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings**, 2015.

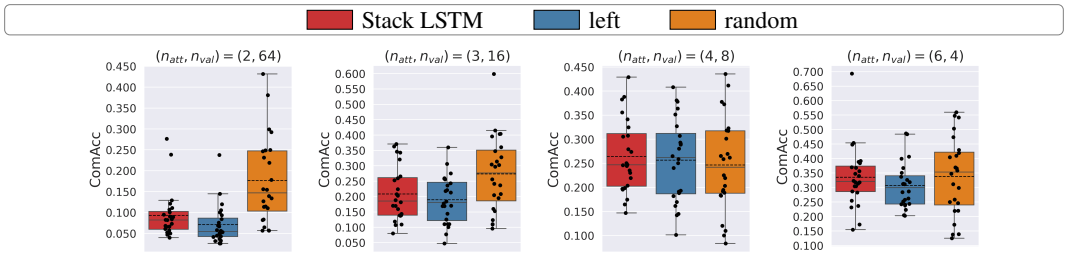


図 3: 予備実験の ComAcc に関する結果

A　実験の詳細設定

A.1　実験 I

メッセージ空間に関して、メッセージの長さの最大長は 8、メッセージを構成するシンボルの種類は EOS を含めて 4 とする。アーキテクチャに関するパラメタとして、sender と receiver の隠れベクトルは 512 次元、Neural Stack に積まれるベクトルも 512 次元とし、シンボルの埋め込みベクトルのサイズは 32 次元とする。pop、push、read の強さの最大値 k_u, k_d, k_r は全て 2 とする。最適化には Adam [17] を用い、学習率と L2 正則化項の係数はどちらも 0.0001 とする。また、sender のエントロピー正則化項の係数は 0.5 とする。意味空間は、9:1 の比率で分割され、それぞれ学習データ、テストデータとして用いられる。1 つのイテレーションに対するバッチサイズは 8192 とし、1 つの実行の中で 15000 イテレーションの学習が行われる。

A.2　実験 II

基本的に A.1 節 と同じ設定を用いる。KL ダイバージェンスにかかる係数 β の、REWO アルゴリズム [16] における初期値は 0.001 とする。1 つの実行の中で 10000 イテレーションの学習が行われる。

B　予備実験：Stack LSTM による先行研究 [9] の再現

B.1　実験手法

基本的に A.1 節 と同じ設定を用いる。試した意味空間の設定は $(n_{\text{att}}, n_{\text{val}}) = (2, 64), (3, 16), (4, 8), (6, 4)$ である。1 つの実行の中で 5000 イテレーションの学習が行われる。

B.2　実験結果

図 3 の通り、先行研究 [9] と概ね同様に、random-branching の ComAcc が高くなる結果となった。メッセージの階層構造を予測する際にランダム性が入る場合に ComAcc が高くなるという現象は、特定のモデルによるものではなく、そのランダム性自体が ComAcc に影響を与えていると考えられる。