

RL-SPINN を用いた創発言語の汎化性能の評価

加藤大地¹ 上田亮¹ 宮尾祐介¹

¹ 東京大学

{daichi5967, ryoryoueda, yusuke}@is.s.u-tokyo.ac.jp

概要

ニューラルネットワークで構成されたエージェント同士に会話の模倣をさせ、人工的な言語を創発させて、人間の自然言語と比較・解析する**言語創発**という分野が、近年注目を浴びている。しかし、エージェントのアーキテクチャとして、系列の順序に沿って厳密に左から右に文を読む標準的な RNN を採用することが多く、文の中の隠れた木構造を見出しながら文意を理解する人間の理解プロセスとは大きく乖離している。そこで我々は、強化学習を用いて教師なしで文の木構造を学習できる RL-SPINN をエージェントに組み込み、言語を創発させるモデルを提唱する。実験の結果、RL-SPINN を用いた創発言語は、既存のアーキテクチャを含む複数のベースラインに勝る汎化性能を持つことが確認された。

1 はじめに

人間は日々、意志疎通や情報伝達のために無意識のうちに言語を用いているが、言語自体がどのように創発し発展してきたのかという問いは、未だ謎に包まれたままである。この問いに対する突破口と成り得るものとして、最近注目を集めている研究分野に**言語創発** [1] がある。ニューラルネットワークで構成されたエージェント同士に、人間の会話のシミュレーションを行わせ、その間に生じたメッセージのプロトコルを言語とみなす。この言語が、どのような側面で人間の自然言語に似ている・異なっているのかを調査し、創発した言語が自然言語に似た特徴を持つようになるための条件を探っていく。このような構成論的なアプローチでもって言語の発展の歴史を解き明かしていくことが、言語創発分野における大きな研究テーマの一つである。しかし現状の言語創発の研究においては、エージェントのアーキテクチャとして非常に簡易化されたものを採用することが多い。この問題に対処する研究の一例として、Ri ら [2] は、人間の言語認知プロセスを反映し

たアーキテクチャを含んだエージェントでの実験が少ないことを指摘し、人間の言語理解に重要とされる「注意機構」をエージェントに組み込んで既存のアーキテクチャとの比較評価を行った。

我々も Ri ら [2] と同様に、人間の言語理解のプロセスとエージェントのアーキテクチャとのギャップに問題意識を持った。その中で、言語創発分野における過去の研究の多くが、エージェントのアーキテクチャとして標準的な RNN を採用していること [3, 4] に注目した。人間はメッセージを受け取った際、**系列ベース**の理解 (= 厳密な順番で最初のトークンから最後のトークンに沿った理解) をするわけではなく、**木ベース**の理解 (= 文の構成要素を再帰的に合成しながら最終的な意味を作り上げるような理解) をするとされている [5]。これに対し、RNN は系列ベースに文の意味表現を構成することから、現状のエージェントによる人間の言語理解の模倣は、明らかに不十分である。

そこで我々は、木構造を教師なしで学習する **RL-SPINN** [6] を創発言語の枠組みに組み込むことを提唱する。言語創発分野において、言語を創発させる枠組みとして、**シグナリングゲーム** [7] が頻繁に用いられる。シグナリングゲームでは、**送信者**と**受信者**という2種類のエージェントが登場する。本研究では、受信者がメッセージから意味表現を構成するアーキテクチャとして RL-SPINN を適用し、それ以外の部分は従来と同じアーキテクチャを採用した。

また本研究では、評価指標として、訓練時には不可視なデータに対する性能である**汎化性能**を用いる。Yogatama ら [6] は、RL-SPINN が、自然言語の文意をより良く捉えられる方向へのバイアスがかかるように木構造を学習し、汎化性能が上昇すること実証した。我々も同様に、RL-SPINN を組み込んだ受信者が、送信者からのメッセージの意味をより良く捉えられるようになる木構造を学習し、メッセージごとに柔軟に意味表現を構成できるようになること

で、汎化性能がより上がるだろうという仮説を立てた。実験の結果、RL-SPINN を組み込んだゲームにおける汎化性能が高くなる傾向があることが示された。今後、RL-SPINN の性能をさらに引き出すための手法の開発や、異なる評価指標による評価が行われることが期待される。本研究は、言語創発において、人間の持つ木ベースの理解を模倣するエージェントを用いた初めての研究であり、言語創発分野における新たなトピックの示唆となり得るだろう。

2 背景: LSTM の木ベースな派生

2.1 Tree-LSTM

文の意味表現を、文の統語的な構造に沿って生成する研究は、以前から存在していた [8, 9]。Tai ら [10] も同様に、文意と文の統語的特徴との関係に潜在的価値を見出し、当時複数の NLP タスクで高性能を出していた LSTM [11] を、木ベースなモデル **Tree-LSTM** に一般化することを試みた。標準的な系列ベース LSTM は、左から順にトークンを受け取って意味表現を作り上げていくのに対し、Tree-LSTM は入力として受け取る構文木に沿って意味表現を生成する¹⁾。実験の結果、複数の意味理解タスクにおいて、Tree-LSTM は標準的な系列ベース LSTM を凌ぐ結果を得た。さらに後続の研究においても、木ベースのモデルは系列ベースのモデルに比べて、高い性能を出すことが確かめられている [12, 13]。

2.2 SPINN

Tree-LSTM は、入力としてパースされた結果が文毎に必要なため、外部のパパーザへの依存関係が生じるという問題点があり、使い勝手が悪いことが欠点であった。そこで Bowman ら [14] は、shift-reduce parsing を行うパーザをモデル内部に持ち、文の意味表現を作ると同時に木構造を決定していく **SPINN (Stack-augmented Parser-Interpreter Neural Network)** というモデルを提唱した。

2.3 RL-SPINN

SPINN はテスト時には外部のパパーザを必要としないが、訓練時には構文解析の教師データが必要となる。そこで Yogatama ら [6] は、下流タスクの結果を報酬とした強化学習を用いて、訓練時から教

1) この意味で標準的な系列ベース LSTM は、どんな文に対しても、構文木として完全に left-branching なものを受け取っている Tree-LSTM の特殊系であると言える。

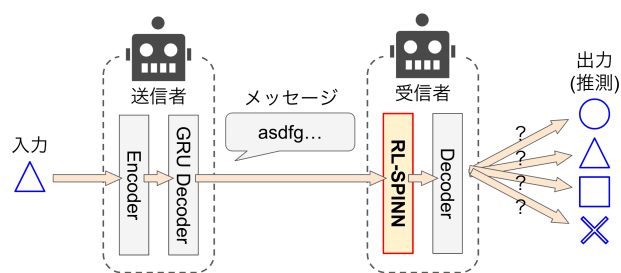


図1 RL-SPINN を組み込んだゲームの簡略図

師なしで木構造を学習する **RL-SPINN (Reinforcement Learning SPINN)** を提唱した。言語創発分野の実験においては、実行毎に言語自体が大きく異なるため、それぞれの言語に対して人間がパースして教師データを作るようなことは当然不可能である。そのため本研究では、教師なしで木構造を学習できる RL-SPINN を採用し、これを受信者に組み込むことで人間の言語理解の仕方を模倣することを試みる。

3 背景: シグナリングゲーム

創発言語を創発させるには、**シグナリングゲーム** [7] という枠組みを基にすることが典型的である。ゲーム内には2種類のエージェント (**送信者・受信者**) がおり、それぞれが別の役割を担う。入力集合を I 、メッセージ集合を M と書くこととする。ゲームは以下のように進行する：

1. ランダムに I からサンプルされた入力 $i \in I$ を、送信者が受け取る。
2. 送信者は入力 i をもとに、メッセージ $m \in M$ を生成し、受信者に渡す。
3. 受信者はメッセージ m のみをもとに、送信者が受け取っていたと思われる入力を推測し、出力 \hat{i} を生成する。
4. i, \hat{i} を比較し、同じであればゲームは成功、そうでなければ失敗とみなす。

上記のゲームを繰り返して学習を進めて行き、正答率が高くなったとき、メッセージ m は送信者・受信者双方にとって何らかの共通意味を持つ記号列であると言える。人間の自然言語が持つ重要な能力「物事を指し示すこと (referring)」ができるものとして、このシンボルを「言語」とみなし、自然言語と比較・解析するのが、創発言語分野における試みである。

4 実験手法

4.1 入力集合・メッセージ集合の設定

入力集合 I として、Chaabouni らの実験 [4] のものと同じの設定を用いる。任意の入力 i は、 i_{att} 個の属性を持ち、それぞれの属性は i_{val} 種類の値を取り得るものとする。つまり、入力集合の大きさは $|I| = i_{\text{att}}^{i_{\text{val}}}$ である。なお実装上、任意の入力 i は、各属性ごとの i_{val} 次元ワンホットベクトルを i_{att} 個連結したものと表現される（つまり入力 i は $i_{\text{dim}} = i_{\text{att}} \times i_{\text{val}}$ 次元のベクトルとして表現される）。

また、有限のアルファベットの集合 V に対し、メッセージ集合 M の要素は V の要素を複数個並べた離散系列で表される。なお、 V の中には EOS として用いる v_{EOS} が含まれており、このアルファベットはメッセージの最後尾にしか登場できないことに注意されたい。厳密には、 V の語彙数 $m_{\text{voc}} (= |V|)$ と、メッセージの最大長 m_{len} が与えられたとき、メッセージ集合 M は以下のように定式化される：

$$M = \bigcup_{l=1}^{m_{\text{len}}-1} \{v_1 \cdots v_{l-1} v_{\text{EOS}} \mid v_i \in V \setminus \{v_{\text{EOS}}\} (i \neq l)\} \\ \cup \{v_1 \cdots v_{m_{\text{len}}-1} v_{m_{\text{len}}} \mid v_i \in V \setminus \{v_{\text{EOS}}\} (i \neq m_{\text{len}}), v_{m_{\text{len}}} \in V\}$$

4.2 提案するシグナリングゲーム

図 1 のような、RL-SPINN を受信者に組み込んだシグナリングゲームを提案する。

送信者のアーキテクチャとして、Chaabouni ら [4] と同一のものを用いる。送信者の最初の層は、入力 i_{dim} 次元のベクトル表現を $n_{S,\text{hidden}}$ 次元の隠れベクトルに変換する線型層である。出力された隠れベクトルは、次の GRU [15] の初期ベクトルとして用いられ、順に 1 つずつアルファベット $v \in V$ を出力していく。なお、訓練時は m_{voc} 個のカテゴリを持つカテゴリカル分布からのサンプリングでアルファベットを選ぶのに対し、テスト時は貪欲に（つまり決定的に）アルファベットを選ぶ。

受信者は、送信者から受け取ったメッセージを使って、送信者が受け取っていた入力を推定する必要がある。まず RL-SPINN によって、メッセージから $n_{R,\text{hidden}}$ 次元の意味表現を生成する。モデルの実装は Bowman ら [14] の論文に準拠する。その後、意味表現を i_{dim} 次元のベクトル表現へと線型層で変換し、これを受信者の推測として出力する。

4.3 ベースライン

本研究では、木構造を教師なしで柔軟に学習することによって生じる RL-SPINN の帰納バイアスを明確にするため、**left-branching**、**right-branching**、**perfect-binary-branching** の 3 つのベースラインを用意する。公平な比較ができるよう、RL-SPINN のうち、木構造の生成を担う部分²⁾のみに手を加えることで、これらのベースラインを実装する。

Left-branching、right-branching は、それぞれ完全に左に寄った木、完全に右に寄った木を決定的に生成するベースラインである。特に、left-branching については、トークンを左から順番に処理していくことになることから、標準的な系列ベース LSTM に対応するベースラインであると言える。また、perfect-binary-branching は、完全に左右対称な木を決定的に生成するベースラインである。

4.4 最適化手法

Chaabouni らの実験 [16, 4] では、送信者を REINFORCE [17] で、受信者を標準的な誤差逆伝播法で学習させる “hybrid” な手法が用いられた。本論文でも同様の手法で学習を行う（詳細は A を参照）。

4.5 各ハイパーパラメタの設定

送信者内、受信者内の隠れベクトルの次元数 $n_{S,\text{hidden}}, n_{R,\text{hidden}}$ はどちらも 500 に設定している。 $(i_{\text{att}}, i_{\text{val}})$ として、(2, 64), (3, 16), (4, 8), (6, 4) を用いる。これらの全ての設定において、入力集合の大きさは同一の $|I| = 4096$ となる。入力全体は、9 : 1 に分割され、それぞれ訓練データ、テストデータとなる。訓練データはエージェントの学習に用いられ、テストデータは汎化性能の評価に用いられる。アルファベットの語彙数 m_{voc} は 4、メッセージの最大長 m_{len} は 8 に設定する。各イテレーションにおけるバッチサイズは 5120 とし、一回の実験の中で 5000 イテレーション行う。各イテレーション毎に、テストデータを用いた汎化性能の評価を行う。

2) 具体的には、RL-SPINN 内の shift-reduce パーザが shift/reduce のアクションを決定する部分を指す。

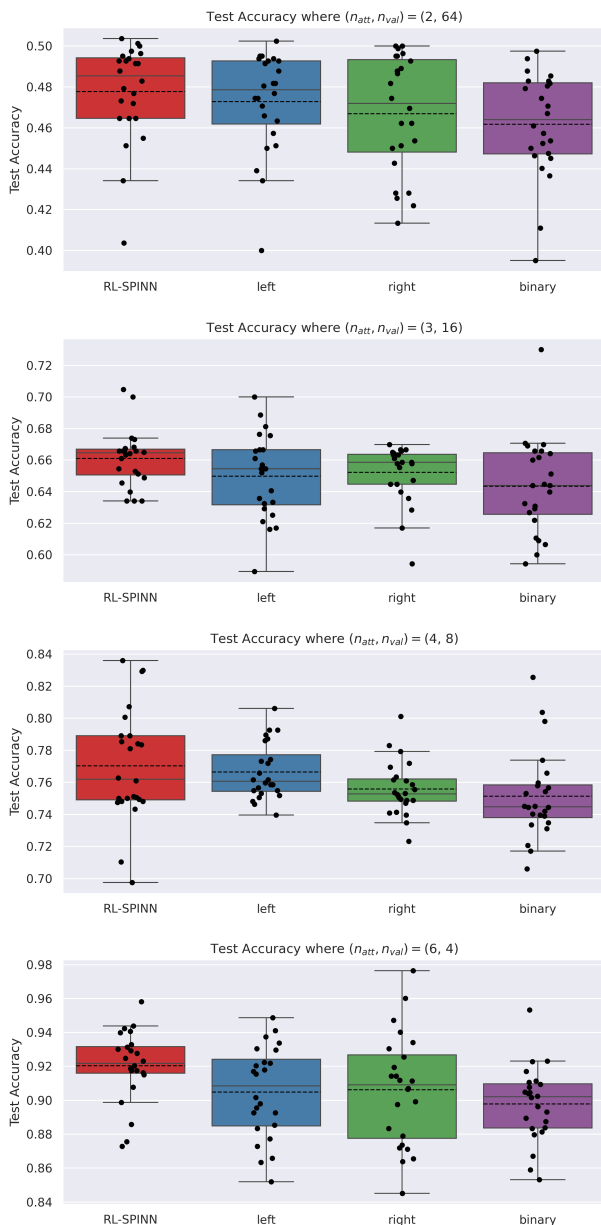


図2 実験における汎化性能の評価結果。4つの図は、上から順に $(n_{att}, n_{val}) = (2, 64), (3, 16), (4, 8), (6, 4)$ の結果である。各図内横軸方向に関して、一番左は RL-SPINN を含む我々のモデルの結果であり、残り3つは左から順にそれぞれ、left-branching、right-branching、perfect-binary-branching ベースラインの結果を表す。縦軸方向はテストデータに対するシグナリングゲームの正解率を示す。箱ひげ図内の点線は、正解率の平均値を表す。

5 実験結果と解釈

第4章で記述した実験設定に従って、同じパラメタに対して random seed のみを変更しながら 24 回の実験を行った。各イテレーションでテストデータに対するシグナリングゲームの正解率を計算し、全てのイテレーションの中で最も高くなった正解率をそ

の実行における汎化性能とした。各実行における汎化性能の値を集め、図2に示す。図からわかる通り、RL-SPINN を含むモデルの汎化性能は、全体的に高くなっている傾向にある。特に、平均値、中央値の観点で結果を見ると、RL-SPINN を含むモデルの汎化性能は、他3つのベースライン全てを凌ぐ結果となった。

しかし、RL-SPINN モデルが学習した木構造を眺める中で、学習された木構造が left-branching や right-branching ベースラインに非常に似た構造になっていることを発見した。そこで、RL-SPINN が学習した木構造の平均の木の深さを計算してみたところ、5.8739³⁾であった。メッセージの長さの平均が 7.1417 であるため、完全に左右のどちらかに寄った場合の平均の木の深さが $7.1417 - 1 = 6.1417$ になることを考慮すると、RL-SPINN が学習した木構造は left-branching や right-branching で使われる木構造に非常に近いと言える。ここから、現状の実験設定では、RL-SPINN の木構造を学習することにより得られる恩恵を、十分には享受できていない可能性が考えられる。

6 おわりに

本論文では、言語内の隠れた木構造を教師なしで学習し、文の意味表現を生成できるモデル RL-SPINN を、シグナリングゲームにおける受信者の中に組み込んだモデルを提唱した。創発した言語の汎化性能を評価する実験を行った結果、我々の RL-SPINN を用いたモデルは、既存の実験設定で頻繁に使われるアーキテクチャを含む複数のベースラインよりも、高い汎化性能を持つ言語を創発することを確認した。

しかし、現状の実験設定では、RL-SPINN の学習する木構造が、ベースラインにかなり近いものになっており、柔軟に木構造を利用できる強みを活かしきれていない可能性が示唆された。また、評価指標は汎化性能のみに限られたため、将来、創発言語の構成性を測る指標として度々用いられる TopSim [18, 19] など、他の指標に関しても、さらなる実験・考察が進められることが期待される。

3) $(n_{att}, n_{val}) = (2, 64)$ の設定で計算、少数第五位を四捨五入。以降の数値も同一の設定である。なお、 (n_{att}, n_{val}) が異なる設定においても同様の傾向を示す結果を得られる。

謝辞

本研究は JSPS 科研費 JP23KJ0768 の助成を受けたものです。本研究を進めるにあたり、有益な助言を頂いた Jason Naradowsky さんに感謝申し上げます。

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] Ryokan Ri, Ryo Ueda, and Jason Naradowsky. Emergent communication with attention. **CoRR**, Vol. abs/2305.10920, , 2023.
- [3] Angeliki Lazaridou, Karl Moritz Hermann, Karl Tuyls, and Stephen Clark. Emergence of linguistic communication from referential games with symbolic and pixel input. In **ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings**. OpenReview.net, 2018.
- [4] Rahma Chaabouni, Eugene Kharitonov, Diane Bouchacourt, Emmanuel Dupoux, and Marco Baroni. Compositionality and generalization in emergent languages. In Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel R. Tetreault, editors, **ACL 2020, Online, July 5-10, 2020**, pp. 4427–4442. Association for Computational Linguistics, 2020.
- [5] David Dowty. Compositionality as an empirical problem.
- [6] Dani Yogatama, Phil Blunsom, Chris Dyer, Edward Grefenstette, and Wang Ling. Learning to compose words into sentences with reinforcement learning. In **ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings**. OpenReview.net, 2017.
- [7] David Lewis. **Convention: A philosophical study**. John Wiley & Sons, 2008.
- [8] Christoph Goller and Andreas Küchler. Learning task-dependent distributed representations by backpropagation through structure. In **Proceedings of International Conference on Neural Networks (ICNN'96), Washington, DC, USA, June 3-6, 1996**, pp. 347–352. IEEE, 1996.
- [9] Richard Socher, Cliff Chiung-Yu Lin, Andrew Y. Ng, and Christopher D. Manning. Parsing natural scenes and natural language with recursive neural networks. In Lise Getoor and Tobias Scheffer, editors, **ICML 2011, Bellevue, Washington, USA, June 28 - July 2, 2011**, pp. 129–136. Omnipress, 2011.
- [10] Kai Sheng Tai, Richard Socher, and Christopher D. Manning. Improved semantic representations from tree-structured long short-term memory networks. In **Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing, ACL 2015, July 26-31, 2015, Beijing, China, Volume 1: Long Papers**, pp. 1556–1566. The Association for Computer Linguistics, 2015.
- [11] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. **Neural Comput.**, Vol. 9, No. 8, pp. 1735–1780, 1997.
- [12] Jiwei Li, Thang Luong, Dan Jurafsky, and Eduard H. Hovy. When are tree structures necessary for deep learning of representations? In Lluís Màrquez, Chris Callison-Burch, Jian Su, Daniele Pighin, and Yuval Marton, editors, **EMNLP 2015, Lisbon, Portugal, September 17-21, 2015**, pp. 2304–2314. The Association for Computational Linguistics, 2015.
- [13] Samuel R. Bowman, Christopher D. Manning, and Christopher Potts. Tree-structured composition in neural networks without tree-structured architectures. In Tarek Richard Besold, Artur S. d’Avila Garcez, Gary F. Marcus, and Risto Miikkulainen, editors, **Proceedings of the NIPS Workshop on Cognitive Computation: Integrating Neural and Symbolic Approaches co-located with the 29th Annual Conference on Neural Information Processing Systems (NIPS 2015), Montreal, Canada, December 11-12, 2015**, Vol. 1583 of **CEUR Workshop Proceedings**. CEUR-WS.org, 2015.
- [14] Samuel R. Bowman, Jon Gauthier, Abhinav Rastogi, Raghav Gupta, Christopher D. Manning, and Christopher Potts. A fast unified model for parsing and sentence understanding. In **ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 1: Long Papers**. The Association for Computer Linguistics, 2016.
- [15] Kyunghyun Cho, Bart van Merriënboer, Çağlar 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, **EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL**, pp. 1724–1734. ACL, 2014.
- [16] Rahma Chaabouni, Eugene Kharitonov, Emmanuel Dupoux, and Marco Baroni. Anti-efficient encoding in emergent communication. 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. 6290–6300, 2019.
- [17] Ronald J. Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. **Mach. Learn.**, Vol. 8, pp. 229–256, 1992.
- [18] Henry Brighton and Simon Kirby. Understanding linguistic evolution by visualizing the emergence of topographic mappings. **Artif. Life**, Vol. 12, No. 2, pp. 229–242, 2006.
- [19] Angeliki Lazaridou, Karl Moritz Hermann, Karl Tuyls, and Stephen Clark. Emergence of linguistic communication from referential games with symbolic and pixel input. In **ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings**. OpenReview.net, 2018.
- [20] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In Yoshua Bengio and Yann LeCun, editors, **ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings**, 2015.

A 最適化手法の詳細

最適化すべきパラメタは、送信者のパラメタ ϕ 、受信者のパラメタ θ である。また、我々のモデルに登場する確率分布は、以下の4種類である：

- 入力集合 I から入力 i をサンプルする確率 $q_{\text{input}}(i)$
- 送信者が i をもとに、メッセージ集合 M からメッセージ m を生成する確率 $S_{\phi}(m | i)$
- RL-SPINN 内部の shift-reduce パーザが、 m をもとに、shift-reduce のアクションの系列 a を生成する確率 $P_{\theta}(a | m)$
- 受信者が m, a をもとに、入力集合 I から推測出力 \hat{i} を生成する確率 $R_{\theta}(i | m, a)$

この状況のもと、我々は以下のような、入力 i と受信者の推測出力に関する確率分布の交差エントロピー誤差の期待値を最小化したい：

$$\mathbb{E}_{i \sim q_{\text{input}}(i), m \sim S_{\phi}(m|i), a \sim P_{\theta}(a|m)} [\mathcal{L}(i, R_{\theta}(i | m, a))] \quad (1)$$

さてここで、やや天下りの的だが、代理関数 f を次のように定義する：

$$f := \mathcal{L}(i, R_{\theta}(i | m, a)) + (\{\mathcal{L}(i, R_{\theta}(i | m, a))\} - b) \log(S_{\phi}(m | i)P_{\theta}(a | m))$$

ここで、 $\{\cdot\}$ は勾配停止記号、 b はベースライン⁴⁾である。なお実験では、ベースラインとして Mean-baseline を用いる。(1) 式の勾配は、 $\nabla_{\phi \cup \theta} E[f]$ に数学的に等しいことが簡単な計算からわかる。本実験では、最適化のためにこの事実を用いて学習を行う。

さらに、メッセージや shift-reduce のアクションの生成における探索範囲を広範にし、学習を促進するため、エントロピーを高く保つようになる正規化項を追加している。本実験においては、送信者、受信者のエントロピー正則化項にかかる係数はそれぞれ、0.5, 0.01 に設定している。

最適化アルゴリズムには Adam [20] を用いる。本実験では、学習率は 0.0001、L2 正則化項の係数も 0.0001 に設定している。

4) ここでのベースラインは、4.3 節での意味ではなく、強化学習分野の文脈でのベースラインである。