タスク指向対話システムの方策学習への Decision Transformer の適用

戸田隆道 [1]　森村哲郎 [2]　阿部拳之 [2]
[1] 株式会社 AI Shift　[2] 株式会社サイバーエージェント
{toda_takamichi,morimura_tetsuro,abe_kenshi}@cyberagent.co.jp

概要

パイプライン型のタスク指向対話システムにおいて, 行動決定を行う Policy モジュールは設計が非常に複雑でコストがかかる. シミュレーターを用いてオンライン強化学習で方策を学習させる取り組みもあるが, 学習のためのシミュレーターの実装など多くの課題が残されている. 我々は Policy モジュールの方策の学習にシミュレーターを必要としないオフライン強化学習の手法である Decision Transformer を適用する. 本稿では Decision Transformer を方策の学習に適用する際の報酬設計について, MultiWOZ 2.1 のデータセットを使い, 学習時と推論時両方で評価を行う.

1　はじめに

パイプライン型のタスク指向対話システムにおけるモジュールの一つである行動決定を行う Policy モジュールは, 現在の対話状態に基づいてシステムが次に行う対話行動を決定する働きをする. Policy モージュールの方策 (以下方策) の設計は対話のストーリーを設計することと同義と言え, 非常に多くの工数がかかる. そのため工数をかけずデータから方策を設計するために, 近年強化学習を使った試みが多くなされている [1][2].

一般的な強化学習 (オンライン強化学習) ではエージェントは環境との相互作用によって学習を進める. 対話方策の学習においては, 人間が直接学習に関わるのは困難であるため, 多くの場合対話シミュレータを使って学習を行っている. しかしシミュレータは複雑な人間の行動を反映させる必要があるため, シミュレータを作ること自体に多大な工数がかかってしまうという問題を抱えている. 加えて, シミュレータに欠陥があると適切でない学習が行われてしまう恐れもある. そこで, 我々はシミュレータを使わないオフライン強化学習 [3] である Decision Transformer[4] を方策の学習に適用する. オフライン強化学習には弊社で運用している AI Messenger Chatbot[1)] のようなチャットボットと有人チャットを組み合わせた対話システムにおいて, 実環境で溜まったデータをそのまま学習に使うことができるといった利点もある.

オンライン, オフライン問わず強化学習には報酬の設計という共通の課題がある. Decision Transformer はそれに加え, 推論時にも報酬をモデルに与える必要がある. [4] で実験されていた Atari などの環境はゲームのスコアなどから現在の状態に対する報酬が容易に定義できるが, 対話においては現在の状態がタスク達成に対してどの程度良いかを定義することは困難である. 本研究では学習時と推論時の報酬の与え方のパターンを試した結果を共有する.

本研究の貢献は以下の 2 点である.

- タスク指向対話の方策の学習に Decision Transformer を初めて適用した
- 方策の学習と推論における Decision Transformer への報酬の与え方の検証を行なった

2　関連研究

本研究に関わる重要な手法に [4] で提案された Decision Transformer がある. 一般的な強化学習は, 環境においてエージェントが試行錯誤することで方策を改善するが, オフライン強化学習手法の一種である Decision Transformer は, 事前に用意されたデータセットからの学習だけで, 実環境と作用せずに良い方策を獲得することを目的としている. オフライン強化学習は実環境での失敗が許容できないドメインや試行錯誤のコストが高いドメインでの活用が期待されている. Decision Transformer はモデルのアーキ

1) https://www.ai-messenger.jp/

テクチャに GPT をベースとしており, 非マルコフ性の問題, つまり将来が現在の状態だけでなく履歴全体に依存してしまう問題を Transformer の Attention の仕組みで効率よく解くことができるといわれている.

方策の学習に Transformer アーキテクチャを利用する手法として [5] が挙げられる. こちらは異なるドメイン間の転移を行う継続的な強化学習を行う際の状態表現として事前に学習された RoBERTa をベースとして利用しているが, 本研究では状態はバイナリベクトルで離散的に扱う.

その他対話に特化した方策の学習方法として [1] が挙げられる. こちらは逆強化学習 [6] に基づいて, より人間の対話に近くなるような報酬関数を求める手法となっている. オンライン強化学習の手法であり, 学習には対話シミュレーターが必要になる点が本稿で扱う Decision Transformer と異なっているが, 将来的には Decision Transformer で学習済みのモデルをオンラインに適用する際などに組み合わせることが考えられる.

3 実験

3.1 比較手法

Decision Transformer は, 時刻 t の状態 s_t, とった行動 a_t, 得られた報酬 r_t の 3 種類の情報をベクトル化して連結したものを 1 トークンとして, 次の行動 a_{t+1} を予測するように学習する. 学習時は行動軌跡が全て入力され, mini-batch 内で長さの異なるものは padding される. 本研究では Decision Transformer の入力の一つである報酬の与え方の検証を行う. ここで報酬は即時的に与えられる即時報酬と, 将来的に得られる即時報酬の累積値（return-to-go) の 2 種類に分けられる. return-to-go は式 1 のように定義される.

$$\widehat{R}_t = \sum_{t'=t}^{T} r_{t'} \tag{1}$$

ここで T は対話における全ターン数, $r_{t'}$ は t' ターン目の即時報酬を示している. 状態 s_t, 行動 a_t と共に return-to-go$\widehat{R}_t$ がモデルに与えられる. 具体的には, 各ターン t でモデルは式 2 で定義される行動軌跡 τ_t を入力として受け取り, 行動 a_t を出力する.

$$\tau_t = \left(\widehat{R}_1, s_1, a_1, \widehat{R}_2, s_2, a_2, \ldots, \widehat{R}_t, s_t\right). \tag{2}$$

ここで注目すべきは, 軌跡 τ に即時報酬ではなく, 今後得られる即時報酬の累積値 return-to-go を用いていることである, この操作により, Decision Transformer は直近の報酬だけでなく, 未来の報酬も考慮して, 行動を選択できるようになる. そのため, 推論時に大きい所望の return-to-go を設定すれば, 長期的に報酬を大きくするような行動を選択することが期待される. 本稿では以下より return-to-go を報酬と呼び, 学習時の報酬を学習時報酬, 推論時の報酬を推論時報酬と定義する. これらを踏まえて, 検証する報酬の与え方を以下に示す.

1. **baseline**
   対話中は学習時報酬を与えず, 対話が成功した時のみ大きな正の学習時報酬を与える. これは一般的な強化学習でよく使われる報酬パターンで, 今回は対話成功時は 40 を与えるように設定する. 推論時報酬は常に 0 とする.
2. **constant -1**
   対話中は常に小さな負の学習時報酬を与え, 対話が成功した時のみ大きな正の学習時報酬を与える. 常に負の学習時報酬が与えられるため, できるだけ早く目的を達成させるような学習がされることを期待している. 今回は対話中は-1, 対話成功時は 40 を与えるように設定する. 推論時報酬は常に-1 とする.
3. **linear**
   対話開始時の学習時報酬を 0, 完了時の学習時報酬を 40 とし, 中間の対話は線形増加させる. 例えば 5 ターンで完了する対話の場合は 0, 10, 20, 30, 40 と増加する. 尚, 小数になった場合は整数に丸め込む. 推論時報酬について, 1. baseline や 2. constant -1 は一定の報酬を与えればよかったが, 本手法は推論時報酬が変動するため, 何ターン目に最大となるかを決める必要がある. 著者が知る限りでは, 終了ターンがわからない状態での推論時報酬の設計が行われている先行研究は無いため 1, 2, 4, 8, 16 の範囲で探索する.
4. **quadratic**
   考え方としては 3. linear と同様であるが, こちらは二次関数的に学習時報酬を増加させる. 例えば 5 ターンで完了する対話の場合は 0, 6, 15, 26, 40 と増加する. 3. linear と同様, 推論報酬の最大値を何ターン目に設定するかを 1, 2, 4, 8, 16 の範囲で探索する.
5. **log**

考え方としては 3. linear と同様であるが, こちらは底を 2 とする対数関数的に学習時報酬を増加させる. 例えば 5 ターンで完了する対話の場合は 0, 17, 27, 34, 40 と増加する. 3. linear と同様, 推論報酬の最大値を何ターン目に設定するかを 1, 2, 4, 8, 16 の範囲で探索する.

これらの手法に加え, シミュレータを用いない他の方策の学習手法として模倣学習 (Behavior Cloning, 以下 BC) を比較する. BC は状態をバイナリで表現したベクトルから行動をバイナリで表現したベクトルを予測する 2 層の全結合ニューラルネットワークで構成される.

3.2 評価方法

評価には MultiWOZ 2.1[7] を利用する. こちらは旅行サポートサービスを行う店員と顧客とのタスク指向対話コーパスの MultiWOZ[8] のアノテーションミスを修正したり適切で無い対話を除外した, より質の良いタスク指向対話コーパスである. 実験には, タスク指向対話システムの研究開発向けツールキットの ConvLab2[9][10] を用いる. ConvLab2 は NLU(Natural Language Understanding)[11][12] や DST(Dialogue State Tracking)[13] などの Policy 以外のパイプライン型のタスク指向対話システムのモジュールや Agenda ベースのユーザーシミュレータ [14] などを提供している.

パイプライン型のタスク指向対話システムは Policy, NLU, DST, に加えて NLG（Natural Language Generation)[15] の 4 つのモジュールを組み合わせて構成される. 本研究では方策の評価を行うために, 残りのモジュールは以下に固定した.

- **NLU**
  BERT による埋め込み表現から発話文中の各単語が表す意図をクラス分類する JointBERT[16]
- **DST**
  NLU が推定したスロットを用いて人手で作成されたルールに従って対話状態を更新する Rule DST
- **NLG**
  発話テンプレート文の集合から, 方策が予測した行動に合うものを選択する Template NLG

評価指標は方策学習の先行研究 [1][5] で利用されており, ConvLab2 でも利用可能な以下の 2 指標を利用する.

- **Success Rate**
  対話の成功率で, ConvLab-2 では”ユーザー要求がすべて満たされ, かつ予約されたエンティティが制約を満たしている割合”と定義されている. 高いほどユーザーの満足する対話を行うことができていると言える.
- **Average Turn**
  成功した対話の平均ターン数. 小さいほど効率よく対話を行なっていると言える.

システムの評価時には, 異なる 5 種類のランダムシードで 100 対話をシミュレーションし, その平均スコアを用いる.

3.3 結果

報酬の与え方の比較の実験結果を表 1 に示す. linear, quadratic, log は推論時の最大報酬設定ターン (Maximum Reward Setting Turn, 以下 MRST) も分けて評価する. 尚, MultiWoZ 2.1 の学習データの Average Turn は 6.728 であった.

表 1 報酬の与え方の比較

手法	MRST	Success Rate	Average Turn
BC	-	0.376	21.554
baseline	-	0.294	17.164
constant -1	-	0.278	16.626
linear	1	0.402	**16.211**
linear	2	0.404	17.495
linear	4	**0.414**	17.630
linear	8	0.404	17.687
linear	16	0.390	17.067
quadratic	1	0.394	16.463
quadratic	2	0.394	16.463
quadratic	4	0.394	16.463
quadratic	8	0.394	16.463
quadratic	16	0.390	16.989
log	1	0.404	16.946
log	2	0.390	16.384
log	4	**0.414**	17.701
log	8	0.400	17.553
log	16	0.408	17.800

4 考察

4.1 実験結果について

baseline や constant -1 といった成功した時にのみ報酬を与える手法より, linear や quandratic, log といった, 対話の完了に向けて徐々に報酬が増加する手法の方が Success Rate は高いことがわかった. 徐々に報酬が増加する手法の中では linear, log が比較的 Success Rate が高く, 推論時の報酬の与え方は MRST が 4 の時が最も高くなった. この結果は対話の序盤に高い報酬を与えた方がよいことを示していると考えている. ただし MRST=1 や MRST=2 など, 極端に短い場合はわずかに Success Rate が下がってしまう. これは最低限必要な対話ターン数が関係していると想定され, ドメインによって理想的な MRST を調整する必要があると考えている.

Average Turn に関しては各手法の間で大きな差は見られなかったが, 若干 MRST が小さい方が小さくなる傾向が見られた. これは, 推論時に与える報酬が対話完了の目標ターンに影響を与えているのではないかと考えた.

適切な報酬設計を行えた場合は BC よりも良い精度を得ていることから, 単純にデータにある行動を模倣するのではなく, データにある行動からより適切な行動を学習できているといえる. 逆に報酬設計が適切でないと BC と比べて大きく精度が落ちることから, 報酬設計が方策の学習に重要な影響をあたえることがわかる.

4.2 今後の展望

今回推論時の報酬は事前に定義された関数に従って与えたが, ユーザー側の肯定的な発言や予約などのスロット抽出完了フラグなどの対話内容に応じて動的に変化する報酬の与え方も考えられる. またオフライン強化学習の強みとしてオンライン強化学習と組み合わせることができることが挙げられるので, [1] のようなオンライン強化学習手法と組み合わせることで更なる精度向上が期待できる.

方策学習に関するテーマとして今回扱わなかったものの一つに, 異なるドメイン間の転移学習がある. NLU や NLG のように GPT や BERT などの一般的な文章で学習した事前学習モデルを使った手法を適用しづらく, また強化学習の転移学習自体が容易では無いとも言われている [17]. MultiWOZ 2.1 はレストラン予約やタクシー配車など複数のドメインを含んでいるので, この部分を分離して学習し, オフライン強化学習での転移学習, または少量データでのドメイン適用といった方向性での研究も検討している.

MultiWOZ 2.1 は人間同士の対話なので非常に質が高く, ほぼ 100%の割合で完了しているといえる. しかし実際の対話プロダクトでは途中でユーザーが離脱してしまうなど対話が失敗してしまうケースも考えられる. そこでシミュレーター同士で対話させることで, 擬似的に失敗した対話を生成し, 学習データに一定割合で混ぜて精度変化を比較する実験を行なった. 擬似的に生成した失敗した対話を混ぜることで精度が低下することがわかったが, 失敗した対話に対する報酬設計に関して, 明確な答えを得ることができなかったので, この部分に関して掘り下げて研究することも検討している. 尚, 実験結果は Appendix A に記載する.

5 おわりに

本稿では, パイプライン型のタスク指向対話システムにおけるモジュールの一つである方策の学習に, オフライン強化学習手法の Decision Transformer を適用し, 報酬設計の検証を MultiWOZ 2.1 のデータセットを用いて行なった. 学習時は対話の完了に向けて徐々に増加するように報酬を与え, 推論時は 4 ターン目に最大の報酬を設定すると最も Success Rate が高くなった.

今後の取り組みとして, より動的な報酬の与え方やオンライン強化学習との組み合わせ, ドメイン間の転移などを検討している. 加えて, 実際の対話を想定して失敗した対話をデータに混ぜた場合の評価も行なったが, 混合比率が上がるにつれて精度が低下することがわかったが, 失敗した対話に対する報酬設計をどのようにすれば良いかがまだわかっていないので, この点に関しても研究を続ける.

謝辞

本論文の作成にあたりご協力頂きました, 株式会社 AI Shift の友松祐太氏, 杉山雅和氏, 東佑樹氏, 二宮大空氏, 下山翔氏, 株式会社サイバーエージェントの邊土名朝飛氏にこの場を借りて厚く御礼申し上げます.

参考文献

[1] Ziming Li, Sungjin Lee, Baolin Peng, Jinchao Li, Julia Kiseleva, Maarten de Rijke, Shahin Shayandeh, and Jianfeng Gao. Guided dialog policy learning without adversarial learning in the loop. **Findings of EMNLP 2020**, 2020.

[2] Pei-Hao Su, Pawe l Budzianowski, Stefan Ultes, Milica Gašić, and Steve Young. Sample-efficient actor-critic reinforcement learning with supervised data for dialogue management. In **Proceedings of the 18th Annual SIGdial Meeting on Discourse and Dialogue**, pp. 147–157, Saarbrücken, Germany, August 2017. Association for Computational Linguistics.

[3] Sergey Levine, Aviral Kumar, G. Tucker, and Justin Fu. Offline reinforcement learning: Tutorial, review, and perspectives on open problems. **ArXiv**, Vol. abs/2005.01643, , 2020.

[4] Lili Chen, Kevin Lu, Aravind Rajeswaran, Kimin Lee, Aditya Grover, Michael Laskin, Pieter Abbeel, Aravind Srinivas, and Igor Mordatch. Decision transformer: Reinforcement learning via sequence modeling. **arXiv preprint arXiv:2106.01345**, 2021.

[5] Christian Geishauser, Carel van Niekerk, Hsien-chin Lin, Nurul Lubis, Michael Heck, Shutong Feng, and Milica Gašić. Dynamic dialogue policy for continual reinforcement learning. In **Proceedings of the 29th International Conference on Computational Linguistics**, pp. 266–284, Gyeongju, Republic of Korea, October 2022. International Committee on Computational Linguistics.

[6] Brian D Ziebart, Andrew L Maas, J Andrew Bagnell, Anind K Dey, et al. Maximum entropy inverse reinforcement learning. In **Aaai**, Vol. 8, pp. 1433–1438. Chicago, IL, USA, 2008.

[7] Mihail Eric, Rahul Goel, Shachi Paul, Abhishek Sethi, Sanchit Agarwal, Shuyang Gao, Adarsh Kumar, Anuj Goyal, Peter Ku, and Dilek Hakkani-Tur. MultiWOZ 2.1: A consolidated multi-domain dialogue dataset with state corrections and state tracking baselines. In **Proceedings of the Twelfth Language Resources and Evaluation Conference**, pp. 422–428, Marseille, France, May 2020. European Language Resources Association.

[8] Pawe l Budzianowski, Tsung-Hsien Wen, Bo-Hsiang Tseng, Iñigo Casanueva, Stefan Ultes, Osman Ramadan, and Milica Gašić. MultiWOZ - a large-scale multi-domain Wizard-of-Oz dataset for task-oriented dialogue modelling. In **Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing**, pp. 5016–5026, Brussels, Belgium, October-November 2018. Association for Computational Linguistics.

[9] Qi Zhu, Zheng Zhang, Yan Fang, Xiang Li, Ryuichi Takanobu, Jinchao Li, Baolin Peng, Jianfeng Gao, Xiaoyan Zhu, and Minlie Huang. Convlab-2: An open-source toolkit for building, evaluating, and diagnosing dialogue systems. In **Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics**, 2020.

[10] Jiexi Liu, Ryuichi Takanobu, Jiaxin Wen, Dazhen Wan, Hongguang Li, Weiran Nie, Cheng Li, Wei Peng, and Minlie Huang. Robustness testing of language understanding in task-oriented dialog. In **Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics**, 2021.

[11] François Mairesse, Milica Gasic, Filip Jurcicek, Simon Keizer, Blaise Thomson, Kai Yu, and Steve Young. Spoken language understanding from unaligned data using discriminative classification models. In **2009 IEEE International Conference on Acoustics, Speech and Signal Processing**, pp. 4749–4752. IEEE, 2009.

[12] Stefan Ultes, Lina M. Rojas Barahona, Pei-Hao Su, David Vandyke, Dongho Kim, Iñigo Casanueva, Pawe l Budzianowski, Nikola Mrkšić, Tsung-Hsien Wen, Milica Gasic, and Steve Young. PyDial: A Multi-domain Statistical Dialogue System Toolkit. In **Proceedings of ACL 2017, System Demonstrations**, pp. 73–78, Vancouver, Canada, July 2017. Association for Computational Linguistics.

[13] Chien-Sheng Wu, Andrea Madotto, Ehsan Hosseini-Asl, Caiming Xiong, Richard Socher, and Pascale Fung. Transferable multi-domain state generator for task-oriented dialogue systems. In **Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)**. Association for Computational Linguistics, 2019.

[14] Jost Schatzmann, Blaise Thomson, Karl Weilhammer, Hui Ye, and Steve Young. Agenda-based user simulation for bootstrapping a POMDP dialogue system. In **Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Companion Volume, Short Papers**, pp. 149–152, Rochester, New York, April 2007. Association for Computational Linguistics.

[15] Tsung-Hsien Wen, Milica Gašić, Nikola Mrkšić, Pei-Hao Su, David Vandyke, and Steve Young. Semantically conditioned LSTM-based natural language generation for spoken dialogue systems. In **Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing**, pp. 1711–1721, Lisbon, Portugal, September 2015. Association for Computational Linguistics.

[16] Wen Wang Qian Chen, Zhu Zhuo. Bert for joint intent classification and slot filling. **ArXiv**, Vol. abs/1902.10909, , 2019.

[17] Matthew E. Taylor and Peter Stone. Transfer learning for reinforcement learning domains: A survey. **J. Mach. Learn. Res.**, Vol. 10, p. 1633–1685, dec 2009.

A　失敗対話を加えた実験

失敗した対話に対しても報酬を設計する必要があるので, 報酬の与え方の比較の実験で最も Success Rate が高く, Average Turn が小さかった linear, MRST=4 を RSO 100%を current best とし, その current best をベースに以下の 6 パターンを試した.

1. **const**
   常に-1 の報酬を与え, 対話終了 (破綻) 時に-40 の報酬を与える
2. **low**
   -40 から 0 まで線形増加させる
3. **middle**
   -80 から-40 まで線形増加させる
4. **high**
   -120 から-80 まで線形増加させる
5. **steep low**
   -80 から 0 まで, 2. low より傾斜をつけて線形増加させる
6. **steep middle**
   -120 から-40 まで, 3. middle より傾斜をつけて線形増加させる
7. **steep high**
   -160 から-80 まで, 4. high より傾斜をつけて線形増加させる

シミュレーターの Policy モジュールは ConvLab2 で提供されている MLE を利用し, 元のデータと 3:1 の割合で混ぜる RSO(Ratio of simulator and original data) 25%, 元のデータと 1:1 の割合で混ぜる RSO 50%, 元のデータと 1:3 の割合で混ぜる RSO 75%の 3 パターンを試した. 結果を表 2 に示す.

表 2　失敗対話を加えた実験

RSO(%)	手法	Success Rate	Average Turn
100	current best	0.414	17.630
75	const	0.286	14.467
75	low	0.292	15.015
75	middle	0.290	14.315
75	high	0.278	14.972
75	steep low	0.282	**14.237**
75	steep middle	**0.294**	15.302
75	steep high	0.290	15.227
50	const	0.232	**11.509**
50	low	0.256	14.766
50	middle	0.254	15.722
50	high	**0.262**	14.921
50	steep low	0.254	14.510
50	steep middle	**0.262**	15.583
50	steep high	0.250	15.706
25	const	0.140	13.182
25	low	**0.142**	14.334
25	middle	0.116	13.980
25	high	0.110	15.574
25	steep low	0.138	**13.140**
25	steep middle	0.116	14.336
25	steep high	0.118	15.643

A.1　考察

失敗対話を加える割合を増やすほど Success Rate も Average Turn も悪化することがわかった. Success Rate は **steep middle** が比較的よい結果を得られたが RSO25%の際は **low** が最もよい結果となった. 一方 Average Turn に関しては何が効果的だったのかわからなかった. この結果に関しては今後検証を進めていきたいと考えている.