生成的後処理ネットワークによる タスク指向型対話システムの最適化

大橋厚元　東中竜一郎
名古屋大学大学院情報学研究科
{ohashi.atsumoto.c0@s.mail, higashinaka@i}.nagoya-u.ac.jp

概要

後処理ネットワーク (post-processing network; PPN) は，対話システム中の各モジュールの出力を事後修正することで，システム全体のタスク達成能力を改善するコンポーネントである．先行研究において，出力が固定次元のモジュールに対する PPN の有効性は示されてきたが，言語生成モジュール (NLG) の出力は後処理できないという制限があった．本研究では，NLG の後処理を実現するため，言語モデルを活用した生成的後処理ネットワーク (GenPPN) を提案する．具体的には，GenPPN をタスク達成に対して強化学習で最適化するため，各発話が全体のタスク達成に与える貢献度を定量化し，報酬関数に導入する．シミュレーション評価実験と人間評価実験を通して，GenPPN による NLG の後処理が，システムの性能改善に有効的であることを確認した[1)]．

1　はじめに

典型的なタスク指向型対話システムは，4 つのモジュール，すなわち，言語理解（NLU），状態追跡（DST），方策（Policy），言語生成（NLG）からなるパイプライン構造を取る [1, 2]．パイプライン型対話システム全体としてのタスク達成能力改善のため，実際の対話の中でこれらモジュールを直接学習する強化学習ベースの手法が，これまで多く提案されている [3, 4, 5]．

近年我々は，学習不可能なモジュール（例えば，ルールベースや API ベース）を含む，任意のモジュールで構成された対話システムの最適化に向けて，後処理ネットワーク（post-processing network; PPN）を提案した [6]．この手法では，モジュール自身を学習する代わりに，モジュールの出力を修正するコンポーネント PPN を強化学習によって最適化する．この PPN は，NLU，DST，Policy の 3 モジュールに適用され，複数の対話システムのタスク達成率を改善した．一方，PPN は多層パーセプトロン（MLP）をベースとした分類モデルによって実装されているため，NLG が出力する自然言語の後処理は行えないという制限があった．

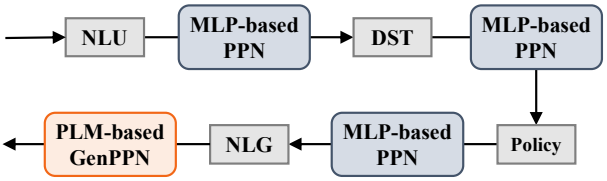

図 1　NLU，DST，Policy のための MLP ベース PPN [6] と，我々が提案する NLG のための PLM ベース GenPPN

本研究では，NLG の後処理を実現するため，事前学習済み言語モデル（PLM）を基盤とした生成的後処理ネットワーク（GenPPN）を提案する．図 1 に，従来の MLP ベース PPN と我々の GenPPN の比較を示す．GenPPN の学習において，PLM の強化学習フレームワーク [7, 8] を適用するには，各ターンにおける発話ごとの報酬（発話レベルの報酬）が必要となる．しかし，タスク指向型対話における一般的な報酬設計では，対話の最後に一度だけ得られるタスク達成／失敗のみが用いられる（対話レベルの報酬）．ここで，対話レベルの報酬を発話レベルに分配できればよいと考えられるが，そのためには，各発話が最終的なタスク達成にどの程度寄与したかが分からなければならない．そこで，本研究では，各発話の意味表現である対話行為 (dialogue act; DA) がタスク達成にどれだけ影響するかを定量化できる **DA 貢献度**を考案し，報酬関数に導入する．

提案手法の有効性検証のため，MultiWOZ データセット [9] とユーザシミュレータを用いた自動評価実験を実施し，異なる NLG が含まれる複数の対話システムのタスク達成能力を，GenPPN が一貫して改善できることを確認する．さらに，ユーザシミュ

1) 本研究で用いたソースコードは https://github.com/nu-dialogue/GenPPN で公開している．

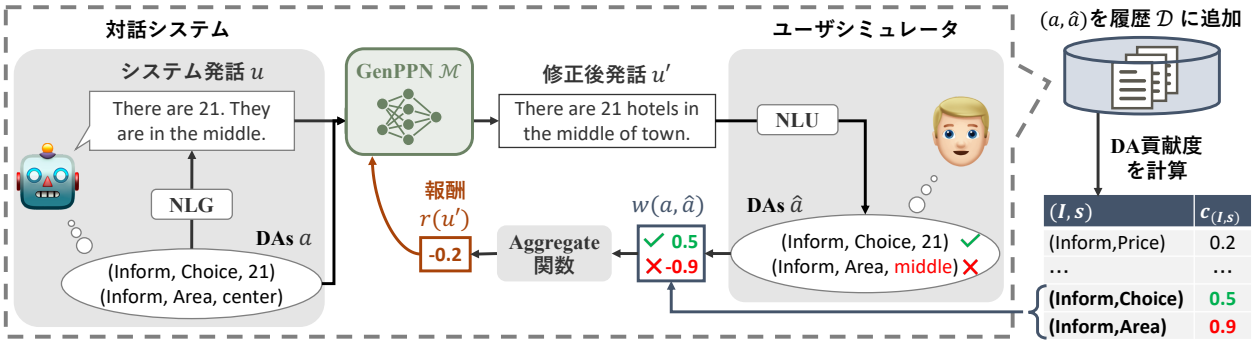

図 2　GenPPN の最適化アルゴリズムの概念図．GenPPN $\mathcal{M}$ は，後処理として，対話システムの NLG によって生成された発話 u を u' に書き換える．この $\mathcal{M}$ による書き換えは，DA 貢献度 $c_{(I,s)}$ に基づいて計算される報酬 $r(u')$ を用いて，システムのタスク達成能力に向けて最適化される．

レータによって最適化された GenPPN が，人間との対話においても有効であることを示す．

2　提案手法

図 2 は，GenPPN とその学習アルゴリズムの概念図を示している．対話システムがユーザとの対話を繰り返す中で，GenPPN は強化学習によって最適化される．各対話において，ターン t におけるユーザ発話から，システムの NLU 及び DST が対話状態を推定する．その対話状態に基づいて Policy が次のシステムの行動を DA a_t として決定する．ここで，DA a は，{(Inform, Choice, 21), (Inform, Area, centre)} のように，意図 I，スロット s，値 v のトリプル 1 つ以上で構成される：

$$a = \{(I_i, s_i, v_i) \mid i = 1, ..., |a|\} \tag{1}$$

次にシステムの NLG は，a_t をシステム発話 u_t に変換する．ここで，GenPPN $\mathcal{M}$ が後処理，すなわち u_t の書き換えを行う．具体的には，対話履歴 h_t，及び u_t と a_t から作成されるプロンプト $x_t = \text{Prompt}(h_t, a_t, u_t)$ を入力された $\mathcal{M}$ が，後処理後の新しい発話 $u'_t \sim \mathcal{M}(x_t)$ を生成する（詳細は 2.1 節を参照）．ユーザは，u'_t を受け取り，そこから理解したシステム DA $\hat{a}_t$ に基づいて次のターン $t+1$ におけるユーザ発話 u_{t+1} を出力する．このやり取りは，ユーザのゴールが達成されるまで，もしくは最大ターン数に達するまで繰り返される．

対話終了後，最大ターン数に達するまでにユーザがタスク達成できたかどうかが，対話結果 $e \in \{0, 1\}$ として評価される（0 / 1 は，成功／失敗を意味する）．本研究の目的は，e の期待値を最大化するような $\mathcal{M}_\theta$ の最適パラメータ θ^* を獲得することである：

$$\theta^* = \arg\max_\theta \mathop{\mathbb{E}}_{U' \sim \mathcal{M}_\theta(X)} [e(U')] \tag{2}$$

なお，$U' = \{u'_1, u'_2, ..., u'_T\}$ は，合計 T ターンからなる対話において，$\mathcal{M}_\theta$ が各ターンのプロンプト $X = \{x_1, x_2, ..., x_T\}$ からサンプルしたシステム発話集合を示す．

ここで，式 (2) は，対話最後に得られる一つの評価値 e のみを用いて，T ターン分の全発話の生成を学習することを示すが，$\mathcal{M}$ の最適化においては，このスパースな報酬は実用的ではない．そこで，各ターンの発話が対話全体のタスク達成にどの程度寄与したかを考慮した発話レベル報酬 r を用いて式 (2) を近似する：

$$\theta^* = \arg\max_\theta \mathop{\mathbb{E}}_{U' \sim \mathcal{M}_\theta(X)} \left[\sum_{u'_t \in U'}^{T} r(u'_t) \right] \tag{3}$$

GenPPN は，r を報酬とした強化学習によって更新される（学習の詳細なアルゴリズムは A.1 を参照されたい）．以降では，GenPPN と r の設計の詳細をそれぞれ説明する．

2.1　PLM による生成的後処理

GenPPN としては，Transformer [10] ベースの PLM を仮定する．入力プロンプトには，対話履歴と DA を考慮して，オリジナルのシステム発話を書き換えるための指示が含まれる．GenPPN は強化学習を通して，システム発話の最適化な書き換え u' を生成するように学習されるが，学習の初期時点であっても最低限プロンプトに従った応答生成ができることが好ましい．そのため，タスク指向型対話 [11, 12] を含む多くの NLP ベンチマークで，高い指示追従能力を示している instruction-tuned model [13, 14] を，GenPPN の基盤モデルとして採用する．

2.2 DA 貢献度による報酬設計

NLG および GenPPN の役割は，ユーザに DA を伝えること，つまり $a = \hat{a}$ となる発話を生成することである．したがって，式 (3) における発話レベル報酬 r としては，まず a と $\hat{a}$ の一致率が考えられる．しかし，この計算には対話レベルの評価値 e が考慮されていないため，式 (2) の近似には不適である．我々は，各発話の a が最終的な e にどれだけ影響したかを定量化するための DA 貢献度を設計し，これを a と $\hat{a}$ の一致率に重み付けすることで，対話レベルな評価値を考慮した発話レベル報酬を設計する．

まず，各対話の終了時，a_t と $\hat{a}_t$，そしてその対話の結果 e からなるトリプルを全ターン分，DA 履歴 $\mathcal{D}$ に追加する．任意の対話数のサンプル後，$\mathcal{D}$ に含まれる全 $(a, \hat{a})$ ペアを，それぞれの e の値，つまりタスク達成／失敗に基づいて，タスク成功 DA 集合 $\mathcal{S}$ と，タスク失敗 DA 集合 $\mathcal{F}$ に分ける：

$$\mathcal{S} = \{(a, \hat{a}) \mid (a, \hat{a}, e) \in D, e = 1\}$$
$$\mathcal{F} = \{(a, \hat{a}) \mid (a, \hat{a}, e) \in D, e = 0\}$$

そして，各 $(I, s, v) \in a$ の貢献度 $c_{(I,s)}$ を以下のように計算する：

$$c_{(I,s)} = \frac{n^{\mathrm{Rec},\mathcal{S}}_{(I,s)} + n^{\mathrm{Unr},\mathcal{F}}_{(I,s)}}{n^{\mathrm{Rec},\mathcal{S}}_{(I,s)} + n^{\mathrm{Rec},\mathcal{F}}_{(I,s)} + n^{\mathrm{Unr},\mathcal{S}}_{(I,s)} + n^{\mathrm{Unr},\mathcal{F}}_{(I,s)}} \tag{4}$$

ここで，$n^{\mathrm{Rec},\mathcal{S}}_{(I,s)}$ は，タスク達成した対話の中で，(I, s) がユーザに正しく伝わった回数である：

$$n^{\mathrm{Rec},\mathcal{S}}_{(I,s)} = \sum_{(a,\hat{a}) \in \mathcal{S}} \sum_{(I',s',v') \in a \cap \hat{a}} [(I, s) = (I', s')]$$

同様に，$n^{\mathrm{Unr},\mathcal{S}}_{(I,s)}$ はタスク達成対話の中で (I, s) がユーザに伝わらなかった回数，$n^{\mathrm{Rec},\mathcal{F}}_{(I,s)}$ はタスク失敗対話の中でユーザに伝わった回数，$n^{\mathrm{Unr},\mathcal{F}}_{(I,s)}$ はタスク失敗対話の中でユーザに伝わらなかった回数を示す．つまり，$c_{(I,s)}$ は，(I, s) がユーザに伝わること（又は，伝わらないこと）とタスク達成（又は，タスク失敗）の共起確率を示しており，これは，(I, s) のタスク達成に対する重要度を定量化していると考えられる．

最後に，$c_{(I,s)}$ を用いて，GenPPN が生成した発話 u'_t の報酬 $r(u'_t)$ を以下のように計算する：

$$r(u'_t) = \mathrm{Aggregate}(w(a_t, \hat{a}_t)) \tag{5}$$

$$w(a_t, \hat{a}_t) = \{\tau \cdot c_{(I,s)} \mid (I, s, v) \in a_t \cap \hat{a}_t\} \cup \{-\tau \cdot c_{(I,s)} \mid (I, s, v) \in a_t \cap \bar{\hat{a}}_t\} \tag{6}$$

つまり $c_{(I,s)}$ に対して，(I, s) がユーザに正しく認識されていれば正のスコア τ が，認識されていなければ負のスコア $-\tau$ が重み付けされる．τ はハイパーパラメータであり，Aggregate 関数は，a_t に対して計算された重み付きスコア集合 $w(a_t, \hat{a}_t)$ をスカラ値にマッピングするための関数である．Aggregate 関数としては以下の二つを用意し，いずれが良いかは実験的に検証する：

mean $w(a_t, \hat{a}_t)$ の平均を出力する．

absmax $w(a_t, \hat{a}_t)$ における絶対値が最大のスコアを採用する．これは，貢献度の高い (I, s) を優先的に学習する戦略である．

3 実験

MultiWOZ データセット [9] に基づいて実装された対話システムとユーザシミュレータを用いて，GenPPN の有効性を評価した．MultiWOZ には旅行案内における店員と顧客の対話が収録されている．

本実験を通して，タスク指向型対話システムの評価用プラットフォームである ConvLab-2 [15] を使用した．以降で，本実験で使用した対話システム，NLG ベースライン，GenPPN の実装を説明する．

対話システム NLG および GenPPN による発話生成の性能評価においては，それ以外のモジュール（つまり，NLU，DST，Policy）の性能は高いことが望ましい．そこで，ベースとなる対話システムとしては，ConvLab-2 で実装されたもののうち，最も性能の高いモジュールの組み合わせである BERT [16] ベースの NLU，ルールベースの DST，ルールベースの Policy を採用した．

NLG ベースライン 提案手法が，多様な NLG に対して頑強に機能することを実証するため，アーキテクチャの異なる 4 種類の NLG モデルそれぞれに GenPPN を適用して有効性を検証した：(1) Template NLG，(2) SC-LSTM [17]，(3) SC-GPT [18]，(4) GPT-2 + RL [19]．各モデルの詳細は A.2 を参照されたい．

GenPPN GenPPN のための instruction-tuned PLM としては，Alpaca 7B [20] を採用した．このモデルは，52K からなる instruction データセットによって fine-tuning された LLaMA-7B [21] である．計算コスト削減のため，GenPPN の最適化では LoRA [22] を採用し，Alpaca の self-attention モジュールに追加された少量のパラメータのみを学習した．その他の学習の詳細は，A.3 節を参照されたい．

表 1　各 NLG とそれらに GenPPN を適用した場合の対話システムの評価結果．GenPPN の添字は学習時に用いられた Aggregate 関数を示す．NLG と GenPPN のペアごとに，最も高いスコアが**太字**で表されている．$*$ と $**$ は，McNemar 検定において，それぞれ GenPPN による Success の改善が $p < 0.05$ と < 0.01 で有意であったことを示す．

NLG	Success	Inform	Book	Turn ↓	DA F1
Template NLG	77.25	78.44	83.91	7.67	71.73
+ GenPPN$_{\text{mean}}$	77.93	79.75	84.33	7.63	76.98
+ GenPPN$_{\text{absmax}}$	**78.91***	**79.86**	**85.19**	**7.02**	**78.23**
SC-LSTM	54.00	67.45	67.69	11.65	60.56
+ GenPPN$_{\text{mean}}$	60.64**	81.01	**78.74**	9.42	**79.80**
+ GenPPN$_{\text{absmax}}$	**72.95****	**79.46**	78.46	**7.21**	79.08
SC-GPT	64.94	**78.06**	56.94	7.80	71.53
+ GenPPN$_{\text{mean}}$	**73.63****	76.54	**82.08**	8.03	73.07
+ GenPPN$_{\text{absmax}}$	73.34**	77.34	80.79	**7.50**	**73.72**
GPT-2 + RL	72.36	76.70	76.81	7.47	**81.17**
+ GenPPN$_{\text{mean}}$	74.02	77.10	79.19	7.54	80.98
+ GenPPN$_{\text{absmax}}$	**75.20****	**78.79**	**79.80**	**7.15**	80.08

3.1　自動評価結果

シミュレータを用いた自動評価では，タスク指向型対話システムの評価尺度として標準的な 4 つ **Turn**（対話終了までに何ターン要したか），**Inform F1**（ユーザから要求された情報を適切に提供できたか）**Book Rate**（ユーザの条件に合った施設を予約できたか），**Task Success**（タスク達成率）を用いた．加えて，発話レベルの評価を行うために，a と $\hat{a}$ の一致率を F1 によって評価する **DA F1** も用いた．

表 1 に評価結果を示す．いずれの NLG についても，GenPPN によって，全てのスコアが改善した．特に SC-LSTM においては，GenPPN$_{\text{absmax}}$ によってタスク達成率が約 19 ポイント向上した．その他の NLG についても，absmax もしくは mean の少なくとも一方で，タスク達成率に有意な改善が見られた．なお，Aggregate 関数としては，mean と比べ absmax を用いた方が，最終スコアは全体的に高性能であった．これは，各発話サンプルについて，全ての DA を平均的に学習するよりも，より重要な DA を優先して学習した方が，最終的なタスク達成能力の向上に効果があることを示している．

なお，GPT-2 + RL においては，GenPPN の適用によって DA F1 が低下していた．これは，GPT-2 + RL は，発話レベル，すなわち DA F1 に最適化されていることを考えると合理的な結果であると言える．そして，この GPT-2 + RL に対しても，GenPPN の適用によってタスク達成率が向上していることから，対話全体のタスク達成能力向上のためには，発話レベルだけでなく対話レベルも考慮した報酬設計が重要であることが示唆された．

表 2　人間評価結果．N は各システムを評価した対話者の数である．“Und.”，“App.”，“Sat.” は，それぞれシステムの理解力，システム応答の適切さ，対話満足度に関する主観評価を示す．$*$ と $**$ は，McNemar 検定において，GenPPN による Success の改善が，$p < 0.05$ と < 0.01 で有意であったことを示す．

NLG	N	Success	Turns	Und.	App.	Sat.
SC-LSTM	54	33.33	17.72	**4.09**	**4.17**	**4.04**
+ GenPPN	50	**52.00***	**15.08****	4.04	4.14	3.98

3.2　人間評価結果

シミュレーションにより最適化された GenPPN が人間ユーザに対しても有効であるかを検証した．ここでは，3.1 節において改善幅が最大であった SC-LSTM + GenPPN$_{absmax}$ の組み合わせを採用した．Amazon Mechanical Turk で合計 100 人以上の対話者を募集した．各話者は SC-LSTM のみを用いたシステム，又は GenPPN が装備されたシステムのいずれかと最大 20 ターン対話しタスク達成を判断した．また，システムの理解能力，システム応答の適切さ，対話の満足度を 5 段階で主観的に評価した．

表 2 に評価結果を示す．GenPPN によって，タスク達成能力に関わる 2 つの尺度，すなわち Success と Turn が有意に改善していることがわかる．一方で，主観評価尺度である，理解力，応答適切さ，満足度には，有意な差は見られなかった．理由として，GenPPN はタスク達成に関する尺度のみを考慮した報酬関数によって最適化されたことが考えられる．発話の自然性など，よりユーザの主観評価を考慮した報酬設計を導入することでこれらのスコアが改善する可能性がある．

4　おわりに

本研究では，タスク指向型パイプライン対話システムのタスク達成能力強化を NLG の後処理によって実現するため，生成的後処理ネットワーク（GenPPN）を提案した．MultiWOZ データセットとユーザシミュレータを用いた実験によって，GenPPN は，異なる NLG で構成される多数の対話システムに頑強に機能することが示された．また，人間評価実験によって，シミュレータに対して最適化された GenPPN が人間に対しても有効であることも実証された．

謝辞

本研究は科研費「モジュール連動に基づく対話システム基盤技術の構築」(課題番号 19H05692) の支援を受けた.

参考文献

[1] Steve Young, Milica Gašić, Blaise Thomson, and Jason D. Williams. Pomdp-based statistical spoken dialog systems: A review. **Proc. IEEE**, pp. 1160–1179, 2013.

[2] Zheng Zhang, Ryuichi Takanobu, Qi Zhu, MinLie Huang, and XiaoYan Zhu. Recent advances and challenges in task-oriented dialog systems. **Science China Technological Sciences**, pp. 1–17, 2020.

[3] Hwaran Lee, Seokhwan Jo, Hyungjun Kim, Sangkeun Jung, and Tae-Yoon Kim. SUMBT+LaRL: Effective Multi-Domain End-to-End Neural Task-Oriented Dialog System. **IEEE Access**, pp. 116133–116146, 2021.

[4] Zichuan Lin, Jing Huang, Bowen Zhou, Xiaodong He, and Tengyu Ma. Joint System-Wise Optimization for Pipeline Goal-Oriented Dialog System. **arXiv preprint arXiv:2106.04835**, 2021.

[5] Zhi Chen, Lu Chen, Xiang Zhou, and Kai Yu. Deep reinforcement learning for on-line dialogue state tracking. In **Proceedings of Man-Machine Speech Communication**, pp. 278–292, 2023.

[6] Atsumoto Ohashi and Ryuichiro Higashinaka. Post-processing networks: Method for optimizing pipeline task-oriented dialogue systems using reinforcement learning. In **Proc. SIGDIAL**, pp. 1–13, 2022.

[7] Daniel M Ziegler, Nisan Stiennon, Jeffrey Wu, Tom B Brown, Alec Radford, Dario Amodei, Paul Christiano, and Geoffrey Irving. Fine-tuning language models from human preferences. **arXiv preprint arXiv:1909.08593**, 2019.

[8] Nisan Stiennon, Long Ouyang, Jeffrey Wu, Daniel Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul F Christiano. Learning to summarize with human feedback. In **Proc. NeurIPS**, pp. 3008–3021, 2020.

[9] 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 **Proc. EMNLP**, pp. 5016–5026, 2018.

[10] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, L ukasz Kaiser, and Illia Polosukhin. Attention is All you Need. In **Proc. NeurIPS**, pp. 5998–6008, 2017.

[11] Wenbo Pan, Qiguang Chen, Xiao Xu, Wanxiang Che, and Libo Qin. A preliminary evaluation of chatgpt for zero-shot dialogue understanding. **arXiv preprint arXiv:2304.04256**, 2023.

[12] Vojtěch Hudeček and Ondrej Dusek. Are large language models all you need for task-oriented dialogue? In **Proc. SIGDIAL**, pp. 216–228, September 2023.

[13] Jason Wei, Maarten Bosma, Vincent Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M. Dai, and Quoc V Le. Finetuned language models are zero-shot learners. In **Proc. ICLR**, 2022.

[14] Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Eric Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, et al. Scaling instruction-finetuned language models. **arXiv preprint arXiv:2210.11416**, 2022.

[15] 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 **Proc. ACL**, pp. 142–149, 2020.

[16] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In **Proc. NAACL-HLT**, pp. 4171–4186, 2019.

[17] 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 **Proc. EMNLP**, pp. 1711–1721, 2015.

[18] Baolin Peng, Chenguang Zhu, Chunyuan Li, Xiujun Li, Jinchao Li, Michael Zeng, and Jianfeng Gao. Few-shot Natural Language Generation for Task-Oriented Dialog. In **Findings of EMNLP**, pp. 172–182, 2020.

[19] Atsumoto Ohashi and Ryuichiro Higashinaka. Adaptive natural language generation for task-oriented dialogue via reinforcement learning". In **Proc. COLING**, pp. 242–252, 2022.

[20] Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. Stanford Alpaca: An Instruction-following LLaMA model. https://github.com/tatsu-lab/stanford_alpaca, 2023.

[21] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. LLaMA: Open and efficient foundation language models. **arXiv preprint arXiv:2302.13971**, 2023.

[22] Edward J Hu, yelong shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In **Proc. ICLR**, 2022.

[23] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal Policy Optimization Algorithms. **arXiv preprint arXiv:1707.06347**, 2017.

[24] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. **OpenAI blog**, Vol. 1, No. 8, p. 9, 2019.

[25] Diederik P Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimization. **arXiv preprint arXiv:1412.6980**, 2014.

A Appendix

A.1 強化学習による GenPPN の最適化

PLM を強化学習により最適化する先行研究 [8] に倣い，Proximal Policy Optimization (PPO) [23] をベースとした目的関数に r（式 (5)）を導入して，GenPPN を学習する．Value network として，スカラ値を出力する線形層（ランダムに初期化されたパラメータ ϕ を持つ）を GenPPN に追加し，全学習可能パラメータを $\psi = [\theta; \phi]$ とする．$\mathcal{M}_\psi$ の確率分布が元の $\mathcal{M}$ から大きく乖離することを避けるため，Kullback-Leibler (KL) divergfence によるペナルティを $r(u'_t)$ に足した値 R_t をターン t における発話 u' の最終的な報酬とする：

$$R_t = r(u'_t) - \beta \log \frac{\mathcal{M}_\psi(u'_t|x_t)}{\mathcal{M}(u'_t|x_t)} \tag{7}$$

この R_t と value network の出力から advantage estimate を計算し，これを用いた clipped surrogate objective $\mathcal{L}(\psi)$ [23] に基づいて GenPPN を最適化する．Algorithm 1 に，学習アルゴリズムの要約を示す．

Algorithm 1 Optimization of GenPPN via PPO

Require: Dialogue system $\mathcal{A}$, User simulator $\mathcal{U}$

Require: GenPPN $\mathcal{M}$

Initialize DA history $\mathcal{D}$ by sampling several dialogues using $\mathcal{A}$ and $\mathcal{U}$
Prepare $\mathcal{M}_{\psi_{\text{old}}}$ with randomly initialized LoRA parameters θ and value network parameters ϕ
for each training iteration **do**
    while #turns does not reach batch size **do**
        Sample a dialogue and $(a_t, \hat{a}_t)$ for each turn t using $\mathcal{A}$, $\mathcal{M}_{\psi_{\text{old}}}$, and $\mathcal{U}$
        Obtain final evaluation result e
        Add $(a_t, \hat{a}_t, e)$ of each t to $\mathcal{D}$
    end while
    Calculate reward R_t using Eq. (7)
    Compute advantage estimates
    Optimize $\mathcal{L}(\psi)$ with a certain epoch and mini-batch size
    Update $\psi_{\text{old}} \leftarrow \psi$
end for

A.2 NLG ベースラインの詳細

Template NLG [15] 各 DA を表す発話の人手によるテンプレート文直接使用する NLG モデル．

SC-LSTM [17] Reading gate 機構を用いた LSTM ベースのモデル．DA のバイナリ表現を入力特徴料とする．

SC-GPT [18] MultiWOZ を含む，7 種類のタスク指向型対話データセットで学習された GPT-2 [24] ベースのモデル．

GPT-2 + RL [19] MultiWOZ に基づいて実装されたユーザの NLU を用い，a と $\hat{a}$ の一致率のみ考慮した報酬で強化学習された GPT-2 ベースのモデル．

A.3 GenPPN の学習詳細

GenPPN の基盤モデルである Stanford Alpaca 7B としては，Huggingface Hub で公開されている学習済み重み[2)] を用いた．入力プロンプトと出力文の最大トークン数は，それぞれ 512 と 128 とした．推論時の生成パラメータは，beam size，temperature，top-p を全て 1 とした．また報酬計算で用いる τ（式 (6)）は 1 に設定した．

表 3 に LoRA 及び PPO におけるハイパーパラメータを示す．強化学習では，合計 200 iteration の学習を行い，1 iteration におけるバッチサイズは 512 ターン（約 50 対話に相当）とした．Adam オプティマイザ [25] を固定学習率 1e-5 で用いた．学習には 16 機の V100 32GB GPU を用い，200 iteration の学習に約 7 時間要した．図 3 に，Template NLG に GenPPN を適用した場合の学習曲線を示す．

表 3 ハイパーパラメータ設定

Hyperparameter Name		Value
LoRA	Target projection matrix of self-attention module	query, key value, output
	Rank	16
	Scaling factor α	16
PPO	Total iterations	200
	Total batch size	512
	Epoch	4
	Total mini-batch size	32
	Learning rate	1e-5
	Optimizer	Adam
	Discount factor γ	1.0
	GAE factor λ	0.95
	Clipping ϵ	0.2
	Coef. of KL penalty β	0.01

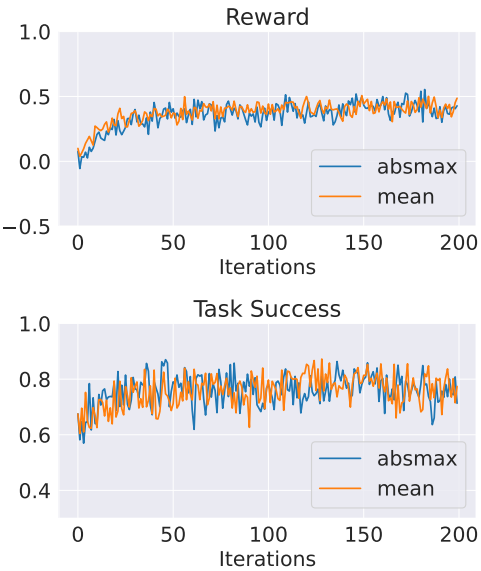

図 3 Template NLG に GenPPN を適用し，それぞれの Aggregate 関数で学習した場合における報酬とタスク達成率の推移

2) https://huggingface.co/tatsu-lab/alpaca-7b-wdiff