ユーザから情報を引き出すための強化学習による応答生成

佃夏野 [1]　品川政太朗 [1,2]　中村 哲 [1,2]
[1] 奈良先端科学技術大学院大学
[2] 理化学研究所革新知能統合研究センター
{tsukuda.natsuno.tn1 sei.shinagawa s-nakamura}@is.naist.jp

概要

雑談対話システムをユーザに長く使用してもらうには，システムがユーザに興味のある話題を提供する必要があるが，まずシステムがユーザの情報を知らねばならない．この時ユーザの心象を損ねず対話でユーザから情報を引き出すのが理想だと考えられ，質問や相槌等の複数の言動を効果的に使い分けることが有望な方法の一つとして考えられる．本研究では初歩として，指定された情報をユーザ発話から抽出する簡単なタスクを設定した．また，タスクを達成するために単純な報酬を用意し，強化学習を利用して既存の対話システムの最適化を行い，システムの応答文の変化について実験を行なった．結果として，システムをタスクに最適化できたことと，的外れな内容の生成への対策の必要性が分かった．

1　はじめに

雑談対話システムを実運用するには，ユーザに長く使用してもらうことは重要な課題の一つである．そのために，システムがユーザの趣味や職業等の情報をユーザから引き出し，ユーザに興味のある話題を提供することが考えられる．ここで情報を引き出すために単に質問のみ繰り返すと，ユーザが煩わしさや不信感を感じ，システムを使用しなくなる恐れがある．これを防ぐには，質問以外に相槌や自己開示等の複数の対話行為をシステムが扱いつつ，情報を引き出せるように戦略的に複数の対話行為を使い分けることが必要となる．また，情報を引き出す戦略を学習するには，正解文と比較して最適化を行う教師あり学習より，タスクの目的に合わせて報酬を設計し，より直接的にタスクを解くように最適化できる強化学習を用いるのが良いと考えられる．理想的にはユーザの心象によって複数の対話行為を使い分けられるように最適化を行うべきであるが，扱う対話行為が多いことやユーザの心象についてのモデリングのしにくさからタスクとして複雑になる．

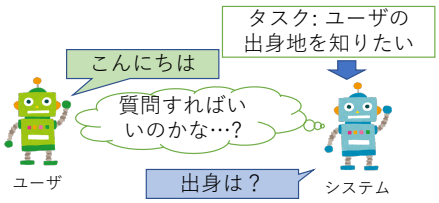

図 1　本研究の対話システムの概要図.

そこで本研究では初歩的な実験として，図 1 のように雑談における情報抽出に目的を限定し，そのための対話戦略を強化学習を用いて学習する．また，対話行為について質問か質問でないかの二つのみを扱うこととし，強化学習の結果として直接的に情報取得をしやすい質問が行われやすくなることを期待する．そこで，「予め欲しい情報を定義し，システムの応答に対し，目的の情報を含む発話をユーザが行ったか」という単純なタスクと報酬を定義し，システムを REINFORCE [1] で最適化する．本来はユーザの心象と相関の高い機械的な指標を用意し報酬に含めるべきであるが，今回は情報の取得のみを報酬として定義する．これによりユーザのシステムに対する心象や会話に対する反応を考慮しないため，対話行為として質問を行いやすくなると考えられる．この際，JPersonChat データセット [2] で予め fine-tuning した Encoder-Decoder モデルを学習対象のシステムと，ユーザの代わりであるユーザモデルの双方に適用し，両者を会話させて学習のためのデータセットを収集する．また，強化学習を適用する前後のシステム応答を比較し，報酬の変化，質問の生成しやすさ，及び生成した応答例を評価する．

2　関連研究

2.1　対話による情報抽出

Han らの研究 [3] では，カウンセリング対話における患者から情報を引き出す 5W1H の質問文の作成

表 1　タスク達成判断の例.

取得したい情報	正解の固有表現	ユーザ発話	タスク達成	報酬
私の趣味は [MASK] です	囲碁	私は福岡生まれです	×	0
私の趣味は [MASK] です	囲碁	私もよく囲碁を打ちます.	○	1

と，患者の回答から対応する情報を抽出する手法を提案した．この手法ではあらかじめ用意したテンプレートをもとにした質問文の作成と情報抽出により，テンプレートに沿った質問と発話については自然な質問作成と正確な情報抽出が行えた．しかし，テンプレートにない多様な質問や発話には対応できなかった．Cotris らの対話システム [4] では対話からイベントに関する人や場所，日付等のユーザの情報を抽出・統合し，外部利用が可能なモジュールを作成した．また，Tigunova らの研究 [5] ではユーザ発話の潜在的な特徴量やスピーチスタイルを元に，趣味や職業等のユーザの個人情報の推定を行なった.

2.2　既存研究と本研究の差分

対話からの情報抽出についての既存研究 [3][4][5] では，ユーザ発話の中にユーザ自身の情報が含まれることを前提とし，いかに情報を抽出するかが研究の主眼とされてきた．これに対し本研究では，日常的な会話で考えられるように，会話の運び方や話し相手への心象によりユーザ発話に情報が含まれない場合があることを踏まえ，心情を損ねずユーザが自身の情報について言及するような対話戦略の学習を目標にしている.

3　提案手法

3.1　タスク設定

本研究では，対話システムが行なった応答 R に対して，ユーザが発話 U を返すという設定の下で、ユーザ発話 U にユーザ情報 I が含まれるように，応答 R をシステムが生成することを目標にする．ここで，取得したいユーザの情報を，今回は職業や出身地といった固有表現として抽出可能なものと定義する．また，この固有表現は，ユーザから取得可能で，かつ，正誤判定のために正解の固有表現をデータセット中に保持できるものとする．タスクが達成できたかどうか判断する変数 $isSuccess$ はシステムの応答に対するユーザの発話から判断され，式 1 となる.

$$isSuccess = \begin{cases} True & (I \in U) \\ False & (I \notin U) \end{cases} \tag{1}$$

また，固有表現抽出とタスク達成の評価の例を表 1 に示す．まず，このタスクで取得したいユーザの情報は表 1 のように，固有表現にあたる部分が「[MASK]」で置換された文として表現される．次に，ユーザ発話に固有表現が含まれ，かつそれが正解であった場合，タスクについて成功したとみなし，ユーザの発話に正解の固有表現が一切含まれていなかった場合にタスク失敗とみなす．また，ユーザ発話中に複数の固有表現が含まれている場合，一つでも正解の固有表現と合致していればタスクを達成したとみなす.

3.2　対話システムの概要

本研究の対話システムは，ユーザ自身について得たい情報とユーザ発話の２つをもとにして応答を生成するモデルからなる．概要を図 2 に示す.

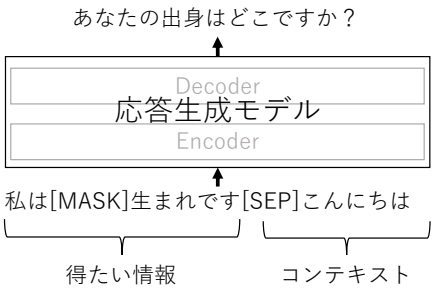

図 2　対話システムの概要.

まず図 1 のようにユーザについて知りたい情報のうち固有表現部分をマスクした文と，コンテキストを繋げた文を区切り文字 [SEP] で繋げ入力文とする．次に，これをトークン化し，Encoder-Decoder モデルからなる応答生成モデルに入力し，最後にシステムの応答を得る．これにより，システムは知りたい情報を引き出すための対話行為と応答の内容，加えて前のターンの会話から大きく外れないことを考慮して，応答を生成できると考えられる.

3.3 学習方法

学習方法には REINFORCE [1] を用い，応答生成モデルを最適化する．まず前提として，システムとユーザモデルを会話させる際に，図 3 のようにユーザモデルには正解となるユーザについての情報「私は福岡出身です」が与えられ，システム側にはその情報のうち固有表現をマスクしたものが与えられる．これは，学習・推論のいずれの場合でも同様である．

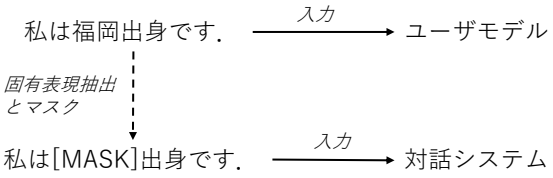

図 3 報酬計算のためのデータ処理とそれぞれのシステムに与えられる入力．

次に報酬 r について，先程の前提を元に，コンテキストを元にした 1 ターンの会話でユーザからマスクした固有表現が得られれば $r = 1$，得られなければ $r = 0$ とする．以上のような学習設定と報酬を元に，予め学習前のシステムとユーザモデルを会話させ，報酬を計算しておくことで強化学習用のデータセットを構築しておく．

次に，用意したデータセットからの学習方法について述べる．今回のような言語生成に対して REINFORCE を適用する場合，損失 L は，入力系列 y のトークン長 T，言語モデル（方策関数）π_θ と時間 t までの状態 s_t に対して，

$$L = -\frac{1}{T}\sum_{t=1}^{T} r \log \pi_\theta(y_t|s_t) \tag{2}$$

となることが知られている．これは，通常の損失 $\log \pi_\theta(y_t|s_t)$ に対して報酬 r で重み付けし，Teacher forcing で学習することに等しい．通常の REINFORCE では，モデル更新にしたがってモデルが新たに生成する文も変化していくが，この方法では一般的に学習が不安定になりやすい．そこで本研究では，強化学習を行う前のシステムとユーザモデル間の対話によって対話のデータセットを予め作成しておき，このデータセットからサンプリングした対話データに対して報酬を評価し、訓練に用いた．

3.4 ユーザモデル

強化学習を行う際に用いるユーザモデルについて説明する．ユーザモデルには，図 4 のように予め与えた固定のペルソナに沿った発話をするように学習した応答生成モデルを用いる．

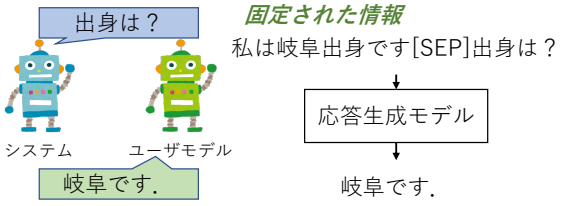

図 4 ユーザモデル概要図．

このユーザモデルでは，まずユーザについて固定された情報とコンテキストを連結した文が，入力文として Encoder-Decoder モデルに与えられる．学習時には入力文に対応したデータセット中の返答文を教師データとしてユーザモデルの学習を行う．推論時には，与えられた情報とコンテキストに矛盾しない発話が生成される．

4 実験設定

4.1 データセット

まず，対話データを作成する際に用いる，対話システムとユーザモデルの fine-tuning に使用する JPersonaChat [2] について述べる．JPersonaChat は PersonaChat [6] の設定をもとにして，クラウドワーカーが与えられたペルソナに沿ってお互いを深く知り合うよう 2 者間で話した対話を収集したものである．各会話のそれぞれの話者にペルソナと呼ばれるユーザに関する情報が「私は横浜出身です」「私は島根県在住です」といった文の形で付与されている．このうち，対話データのみを用いてシステムの生成モデルを fine-tuning する．またユーザモデルは，対話データに加えてそれぞれの対話に結びついた話者のペルソナのうち，固有表現を含むペルソナのみを一つ抽出し，3.4 で示した方法で fine-tuning を行う．これにより，与えた 1 つのペルソナに矛盾しない発話をユーザモデルに行わせる．

次に，強化学習用のデータセットについて述べる．fine-tuning を行ったシステムとユーザモデルについて，ユーザモデルのペルソナを変更しつつ 20 ターンの会話を行い，過去 3 発話程度のコンテキスト，システム応答，ユーザ発話，およびユーザ発話

表 2 学習した対話システムによる応答生成例と得られた報酬．テストセットや生成例に含まれる正解の情報，コンテキスト及びユーザモデルの発話については表に収まらないため省略した．

システムの応答	報酬
そうなのですね。私は、最近カラオケが好きでよく行きます。お休みの時は何をされるんですか。	0
私はテレビ業界で働いています。あなたは、何のお仕事をされているんですか?	1
はい、大丈夫ですよ。私はキャリアコンサルタントをしています。	1

から計算される報酬の 4 つからなる対話データセット作成した．このデータセットには，227 個のペルソナ，227 個の対話，および 9080 発話が含まれており，データセットの総数は 4540 個である．このうち 3971 個を学習セットとし，569 個をテストセットとして評価に用いる．

4.2 評価尺度

評価尺度として，テストセットで得られる報酬の総和と，質問を行っている応答の総数を用い，システムが報酬を得やすくなったか，および質問をしやすくなったかについて評価する．また，質問の報酬獲得率を計算し，質問で報酬を獲得できているかについて評価する．質問の報酬獲得率の計算式は

$$\text{質問の報酬獲得率} = \frac{\text{報酬を得られた質問の総数}}{\text{質問をしている応答の総数}} \tag{3}$$

となる．

5 実験結果と考察

4.2 節で述べた尺度による評価の結果を表 3 に示す．学習前後の報酬の総和を比較すると，学習後のシステムの方が多くなっており，また，質問の数も多くなっていることから，狙った方向性にシステムが正常に学習できていると考えられる．しかし，学習した後は質問が 2 倍近く増加しているのに対して，recall はあまり増加していない．この結果から，質問という対話行為を選択しやすくなったものの，知りたいことを適切に聞けるような発話ができてはいないと考えられる．

表 3 報酬の総和と質問数・質問の報酬獲得率の変化による比較．

	報酬の総和	質問の数	質問の報酬獲得率
学習前	29	69	0.159
学習後	43	128	0.172

そこで，詳しく結果を分析するために，テストセットから生成した応答を表 2 に示す．表 2 の上から 1 つ目の例は，質問はしたが報酬に結び付かなかった応答の代表的な例である．報酬に結び付かなかった質問は，このように「お休みの時は何をされるのか」という内容のものが多かった．これは学習セット中で，報酬を獲得できた質問が 570 個であるのに対し，これに類する質問が，44 個含まれており，頻出する質問を行うよう学習したためと考えられる．続いて，表 2 上から 2 つ目の例は，報酬を獲得できた質問の代表的なものである．報酬を獲得できた質問は，このように質問の答えのドメインが絞られる内容のものが主になっていた．加えて，表 2 の上から 3 つ目のように，質問でなくても報酬が得られた応答としては，会話のドメインを絞った内容のものが主になっている．これらのことから，今回用いたユーザモデルに対して，システムがドメインを絞った直接的な質問を行うことで，更に報酬を増やすことが期待できると分かった．

6 まとめ

本研究では初歩的な実装として，ユーザから指定された情報を引き出すことのみをタスクとして，ユーザが情報を話したか，という単純な報酬を用いて強化学習で対話システムを最適化し，学習したモデルの応答文の変化について調査を行なった．その結果，質問を生成しやすくなり，タスクへの最適化が行えたことが確認できた．また，ユーザから対話的に情報抽出を行うためには，情報抽出のための適切な内容を含む必要があるという課題も分かった．

本研究の本来の最終的な目標は，対話行為を適切に使い分けることでユーザの心象を損ねず対話でユーザから情報を引き出すことである．今後はユーザが情報を話しやすい状態かの判断を報酬設計に組み込み，質問や相槌等の対話行為をシステムが適切に使い分けることを目指す．また，今回の課題も踏まえて対話行為だけでなく，応答の内容についても適切なものを生成できる機構をシステムに導入することを検討していく．将来的には，取得すべき情報を増やしてタスクを複雑化し，本研究を拡張してマルチターンの対話による最適化を行い，最終的に人手の評価を行う予定である．

参考文献

[1] Ronald J. Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. **Mach. Learn.**, Vol. 8, No. 3–4, p. 229–256, may 1992.

[2] Hiroaki Sugiyama, Masahiro Mizukami, Tsunehiro Arimoto, Hiromi Narimatsu, Yuya Chiba, Hideharu Nakajima, and Toyomi Meguro. Empirical analysis of training strategies of transformer-based japanese chit-chat systems. **arXiv:2109.05217**, 2021.

[3] Sangdo Han, Kyusong Lee, Donghyeon Lee, and Gary Geunbae Lee. Counseling dialog system with 5W1H extraction. In **Proceedings of the SIGDIAL 2013 Conference**, pp. 349–353, Metz, France, August 2013. Association for Computational Linguistics.

[4] Keith Cortis and Charlie Abela. Semchat: Extracting personal information from chat conversations. In **EKAW 2010• Workshop W2**, p. 14. Citeseer.

[5] Anna Tigunova. Extracting personal information from conversations. In **Companion Proceedings of the Web Conference 2020**, WWW '20, p. 284–288, New York, NY, USA, 2020. Association for Computing Machinery.

[6] Saizheng Zhang, Emily Dinan, Jack Urbanek, Arthur Szlam, Douwe Kiela, and Jason Weston. Personalizing dialogue agents: I have a dog, do you have pets too? In **Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)**, pp. 2204–2213, Melbourne, Australia, July 2018. Association for Computational Linguistics.

A　実験で使用したモデルについて

節 3.1 で行う固有表現抽出は，オープンソース日本語 NLP ライブラリである Ginza で配布されている ja_ginaza モデル[1)]を用いて，JPersonaChat データセットに対して行う．また，応答生成モデルとして，システムとユーザモデルの両方で NTT による日本語事前学習済み Encoder-Decoder モデル [2] を用いる．

1）　https://github.com/megagonlabs/ginza