学　士　学　位　論　文

題　　目

計算量を考慮した対話行為設計に基づく強化学習による発話選択

指 導 教 員

駒谷 和範 教授

報　告　者

黒田 佑樹

2021 年 2 月 8 日

大阪大学工学部 電子情報工学科

計算量を考慮した対話行為設計に基づく強化学習による発話選択

黒田 佑樹

論文梗概

近年，特定の目標を持たない非タスク指向型対話システムに注目が集まっている．非タスク指向型対話の中にユーザの話の聞き役となり，ユーザの話したいという欲求を満たす傾聴型対話システムがある．傾聴型対話システムでは，システムが話の流れに合わない発話を出力すると，ユーザは気持ちよく対話を続けることができない．

そこで本研究では強化学習を用いて，破綻の生じにくい発話選択の戦略の獲得を目指す．強化学習を用いて発話選択の戦略を獲得するには，対話の状態をどのように表現するか，どのような単位で行動を設計するかが重要となる．

発話選択の戦略獲得に強化学習を用いる際，行動の設計としては大きく 2 つの方針がある．一点目は，最もシンプルに，発話全てを行動とする手法である．しかし，この手法では，用いる発話の数によっては探索すべき空間が増加し，計算量も増加する．二点目は，発話集合を特徴ごとに分類して行動とする方法である．しかし，この方法では，同じクラスに分類された発話のうち，どれが選択されるかをコントロールできず，破綻が生じることがある．実際，発話集合を対話行為に分類して行動とした従来手法では，対話行為選択以降の発話選択がランダムであるため，破綻が生じることが多かった．

本研究では，対話行為の一部のみを詳細化して行動とすることで，計算量を増やしすぎず，破綻の生じる発話の出力を防ぐ．アプローチとしては，対話行為を行動として強化学習を行ったうえで，対話行為内からランダムに発話を選択する従来手法を拡張する．まず，従来手法に基づくシステムにおいて，破綻の生じやすい対話行為の順番を分析した．次に，それに含まれる対話行為を詳細化して，そのまま状態と行動の設計に用いる．これによってより細かい行動が表現できるようになり，対話行為内からランダムに発話が選択されることがなくなる．また，設計した状態と行動の組み合わせに関して，負の報酬を設計する．これによって，ある状態で，破綻が生じる行動の出力を防ぐことができる．これらの状態と行動と報酬の設計によって，対話行為内の発話に優先順位がつけられ，破綻の生じない発話を選択しやすくなる．

実験では，破綻の生じにくさと計算時間の 2 つの観点から提案手法を評価した．比較する手法としては，提案手法，対話行為を行動とした手法，発話全てを行動とした手法の 3 つを用いた．破綻の生じにくさに関する評価実験では，3 つの手法で対話を行い，比較した．結果として，破綻の割合が提案手法では 5%，対話行為を行動とした手法で 17%，発話全てを行動とした手法で 6.5%であった．これにより，提案手法は破綻の生じにくさにおいて，対話行為を行動とした手法よりも優れており，発話全てを行動とした手法と同程度の性能を実現していることが示された．また，計算量に関する評価実験では，3 つの手法に関して，探索空間のサイズに比例するエピソード数で学習を行い，そのときにかかる時間によって評価を行った．結果として，学習を行ったときにかかる時間が提案手法では 2777 秒，発話全てを行動とした手法で 6118 秒であった．これにより，提案手法では，破綻の生じやすさに関して同程度の性能をもつ，発話全てを行動とする手法よりも計算時間を削減できていることが示された．

目 次

第1章　序論

対話システムはタスク指向型と非タスク指向型に大別される．タスク指向型対話システムとは，対話を通して特定の目標を達成するためのシステムのことである．タスク指向型対話システムの例としては，ユーザの情報収集を助けるシステム等 [1] がある．一方，非タスク指向型の例としては，ユーザと対話を続けること自体を目的とした雑談対話システムがある．

非タスク指向型対話システムの中でも，ユーザの話の聞き役となるものを傾聴対話システム [2][3] と呼ぶ．傾聴対話システムはユーザの話したい，聞いてほしいという欲求を満たすことが期待される．

傾聴対話システムではユーザに気持ちよく話してもらうために，話の流れに合った質問をしたり，反応を返したりする必要がある．システムがこれらを行った時，話の流れに合わない，不自然な発話を返してしまうことがある．これを破綻と呼ぶ．対話中に破綻が生じれば，ユーザは対話自体に違和感を感じ，気持ちよく対話を続けることができない．そのため，破綻しない発話を出力することが重要である．

システム発話を出力するためのアプローチには主に，ルールベース，生成ベース，用例ベースの 3 つが存在する [4]．ルールベースとはユーザの入力発話に対するシステム応答をルール形式で記述するアプローチである．生成ベースとは単語や文字レベルからニューラルネットワーク等を用いてシステム応答を生成するアプローチである．用例ベースとは予め用意された発話候補からシステム応答を選択するアプローチである．傾聴対話におけるユーザの発話は多様であり，ルールベースではルールの記述が膨大になるため，適していない．また，生成ベースでは，単語や文字レベルから応答を生成するため，出力したシステム発話そのものが破綻してしまう場合がある．そこで，本研究では，一つの応答を複数の文脈に使え，かつシステム発話そのものが破綻することのない，用例ベースのアプローチに着目する．

対話システムでの適切な応答選択のための戦略はしばしば強化学習で定式化される．強化学習とは機械学習の 1 種で，ある状態である行動をとる価値を，対象のタスクの試行を通して更新していき，最適な方策を学習する手法である．強化学習は試行錯誤を繰り返してより良い方策を学習するという性質上，対話システムに適し

ている．

強化学習は用例ベースの対話システムに用いられることがある．用例ベースを用いた対話システムを強化学習で定式化する場合，最もシンプルな方法として，全ての発話をそのまま行動とするものがある．しかし，この方法では，全体の発話集合の数によっては，探索すべき空間が膨大になってしまう．

これを解決するための方法として，発話候補の集合を行動とする強化学習の設計がある．この方法では，同一行動とみなされる発話候補内の発話は，ある状況において，どれを選んでも適切である必要がある．そのため，特徴ごとに発話を分類して状態や行動の設計に用いる．しかし，集合中からどれを選んでも破綻が生じない発話集合を設計することは困難である．実際に，発話集合を対話行為単位に分類して行動とし，その中からランダムに発話を選ぶ従来手法 [5] では，対話に破綻が生じてしまうことがあった．

本研究では，全ての発話を行動にした手法よりも計算量が少なく，また，発話集合を対話行為単位に分類して行動とした手法よりも破綻の生じにくい対話システムの構築を目指す．そのために，対話行為を行動とし，行動の中から発話をランダムに選ぶ手法 [5] をベースとしたアプローチを提案する．提案手法では，この手法の破綻しやすい対話行為の順番を分析し，そこに現れる対話行為内の発話のみを詳細化して状態や行動の設計に用いる．これにより，計算量が増えすぎることなく，同一対話行為内の発話を個々に選択できるようになり，破綻が減少することが期待される．

以下に本論文の章構成を示す．2 章では強化学習を用いた発話選択手法と，その問題点について述べる．3 章では強化学習を用いた発話選択手法の問題点を解決するための，本研究での提案手法の方針と詳細を述べる．4 章では提案手法を用いた対話を，破綻の生じやすさと計算量の観点から評価を行い，提案手法の問題点解決への有効性を示す．5 章では本研究のまとめと今後の課題について述べる．

第2章　強化学習を用いた発話選択手法とその問題点

本章では 2.1 節で，本研究で扱う強化学習と，その手法の 1 種である Q 学習について，基本的な知識を説明する．2.2 節では強化学習を用いた対話システムに関する研究を述べ，本研究の立ち位置を確認する．2.3 節では，発話集合を特徴ごとに分類して行動とする強化学習の設計の中でも，対話行為を用いた従来手法を紹介する．2.4 節では対話行為を行動とした従来手法の問題点を分析する．

2.1　強化学習とは

強化学習とは機械学習の学習方法の１種である．行動によって報酬が得られる「環境」を与えて，各状態において報酬につながる行動が出力されるようにモデルのパラメータを調整する．学習されたモデルは，状態を入力として，とるべき行動を出力する「方策」とみなせる．学習は何回かの行動を１まとまりとしたエピソード単位で行われる．

強化学習では，マルコフ性に従う環境であるマルコフ決定過程 (MDP) を仮定して学習を行うことが多い．マルコフ性とは，遷移先の状態が直前の状態と行動にのみ依存し，報酬が直前の状態と遷移先に依存するというものである．MDP は以下の 4 つの要素で構成される．

- s: 状態
- a: 行動
- T: 状態と行動を引数に，次の状態と遷移確率を出力する関数
- R: 状態と遷移先を引数に，報酬を出力する関数

MDP を仮定した強化学習は図 2.1 のように表される．

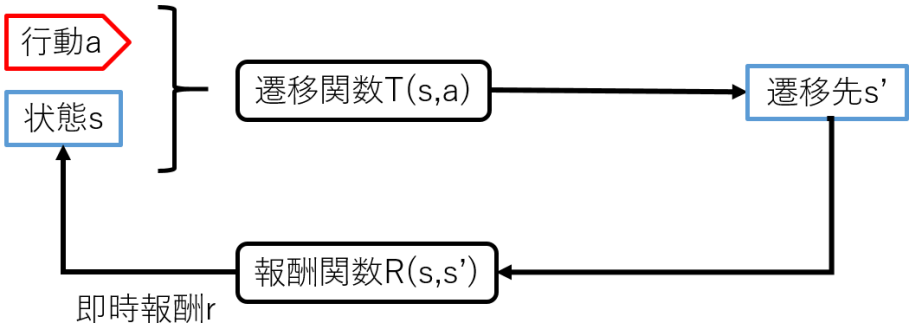

図 2.1 MDP を仮定した強化学習

強化学習にはモデルベースとモデルフリーの手法が存在する．前者は状態遷移関数や報酬関数が既知であり，後者は未知である．後者では遷移関数が分かっていないため，初めはランダムに行動選択を行い，報酬を得ることで，より高い報酬が得られる状態遷移方法を経験的に知る．ある程度の経験が得られたら，これを仮の遷移関数として学習を行う．これを Epsilon-Greedy 法という．

モデルフリーの強化学習として，Q 学習がある．本研究で扱うシステムでは強化学習の手法としてこの Q 学習を用いている．Q 学習ではある状態である行動をとる価値を学習する．この価値を Q 値という．Q 学習の価値更新方法は下記の式 2.1 に従う．

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha(r_{t+1} + \gamma Q(s_{t+1}, s_{t+1}) - Q(s_t, a_t)) \tag{2.1}$$

設定された即時報酬 r_{t+1} と遷移先の価値 $Q(s_{t+1})$ に割引率 γ をかけたものの和によって，その状態行動セットの新たな価値が更新される．更新の速度は学習率 α によって決定される．ある状態である行動をとる価値を記述した表が遷移関数に対応するものであり，Q テーブルと呼ばれる．状態数と行動数の組み合わせの数がこの Q テーブルのサイズとなる．

2.2　強化学習を用いた対話システムの関連研究

本節では強化学習を用いた対話システムに関連する研究を述べ，本研究の位置づけを確認する．

強化学習のシステム行動を応答生成モジュールとした対話システムとして，江頭らの研究 [6] がある．これは「ニュース記事中の一文を表示する」,「Wikipedia 定義文の検索を行う」などの応答生成モジュールを作成し，その選択方策を強化学習に

よって学習するものである．本研究では発話や発話集合そのものを行動として強化学習を行う点が違う．

発話や発話集合そのものを行動とする手法で，最もシンプルなものが，全ての発話を行動とする方法である．しかし，この方法では全体の発話数が多くなった時に，探索すべき空間が爆発的に増えてしまう．佐藤ら [7] は発話単位で行動を設計して強化学習を行い，発話選択の戦略を獲得した．この研究では，深層強化学習を用いることで，探索すべき空間の爆発に対処した．一方，本研究では Q 学習を用い，状態と行動の数を発話分類の工夫によって削減することで，探索空間の増加に対処する．

また，探索空間を増やし過ぎないために考案された方法として，発話集合を何らかの特徴ごとに分割して行動とするものがある．これにより，発話数がいくら増えようが，行動数は分類した数より多くなることはない．Yu ら [8] は発話集合を「ジョーク」や「話題の変更」等に分類して行動設計を行った．本研究では，このような特徴ごとに分類した行動の一部の中身を発話レベルまで詳細化して行動とし，出力の際の優先順位を強化学習することで，意図しない発話の出力を防ぐ．具体的には，2.3 節で詳述する，発話集合を対話行為に分類して行動とする従来手法をベースとする．

2.3 対話行為を単位とする行動設計

2.2 節で述べた，発話全てを行動とした際の探索空間の増加への対応である，発話集合を特徴ごとに分類して行動とする方法の 1 例を詳しく紹介する．西本ら [5] は発話集合を対話行為に分類して行動とする手法を用いた．以下にその手法の状態と行動の設計を示す．

この手法ではシステム発話集合を，表 2.1 に示した 8 つの対話行為に分類したものを行動として扱う．表中の対話行為の記号は表 2.3 に示した用語の組み合わせで表現されている．西本らの研究では，雑談対話コーパス Hazumi1902[9] から話題「スポーツ」「音楽」「食事」「旅行」の 4 つの話題で用いられる発話に，どの話題にも用いられる「default」の発話を加えた発話集合を用いている．表 2.3 の「特定話題」は，前者の 4 つの話題に属する発話で構成される発話集合であることを示しており，「default」は後者の「default」発話に属する発話で構成される発話集合であることを示している．

この手法では状態を以下の 3 つの要素の組み合わせによって表現する．

1. 簡易対話行為 (表 2.2): 4 状態
2. ユーザ発話に特定名詞が含まれるかどうか: 2 状態

表 2.1 対話行為 8 種類

対話行為	説明
qs_o_d	(a) 指示語あり質問 (default)
qs_o_s	(b) 指示語あり質問 (特定話題)
qs_x_s	(c) 指示語なし質問 (特定話題)
re_o_m	(d) 指示語あり応答 (特定話題+default)
re_x_d	(e) 指示語なし応答 (default)
re_x_m	(f) 指示語なし応答 (特定話題+default)
thank	(g) 感謝
io	(h) 情報提供

表 2.2 簡易対話行為

対話行為	説明
qs	質問
re	応答
io	情報提供
ct	話題変更

3. ユーザ心象: 3 状態

簡易対話行為は，発話集合を，行動として設計された 8 つの対話行為よりさらに粗く分類したものである．ユーザ発話に特定名詞が含まれるかという状態は，ユーザ発話に特定名詞が含まれれば指示語あり応答を返したいという思想のもと設計された状態である．また，ユーザ心象はユーザが今現在対話を面白いと思っているかどうかを 3 段階で表している．この値はユーザ入力によって取得されるものとする．これらの組み合わせ 4×2×3 の合計 24 状態がこの手法での状態空間となる．

2.4　対話行為を単位とする行動設計の問題点

対話行為を単位とする行動設計では，システムは対話行為単位でしか発話選択ができない．そのため，対話行為単位での適切な発話選択は可能であるが，発話の具体的な内容までは考慮できず，破綻の生じる発話を選択してしまうことが考えられる．

この手法を実装して，システムと対話を行ったところ，実際にユーザ発話と次のシステム発話がかみ合わない例が見られた．これらを分析した結果，以下の 3 パター

表 2.3 用語説明

用語	説明
qs	質問 (question)
re	応答 (response)
io	情報提供 ()
ct	話題変更 (change theme)
o	指示語あり
x	指示語なし
s	特定話題 (specific)
d	デフォルト話題 (default)
m	特定話題+デフォルト話題 (mix)

ンのような順番で対話行為が選択された時に，破綻が生じやすかった．

【パターン 1】

指示語なし質問

↓

指示語あり質問 (特定話題) or 指示語あり質問 (default)

【パターン 2】

指示語あり質問 (特定話題) or 指示語なし質問

↓

指示語あり応答 or 感謝

【パターン 3】

指示語あり質問 (default)

↓

指示語あり応答 or 感謝

以下で 3 つのパターンについて破綻例を交えながら説明を行う．破綻例の赤字は破綻を表しており，S はシステムの発話であること，U はユーザの発話であることを表している．システム発話に添えられた記号は対話行為を表す．

S1:　スポーツをする目的は何ですか？(qs_x_s)
U1: 健康増進のためですかね
S2(破綻): そのスポーツのおすすめポイントを教えてください (qs_o_s)
U2: どのスポーツ？

図 2.2 パターン 1 の破綻例

パターン 1

パターン 1 の破綻は，システムが指示語なし質問をしたときのユーザの応答に対して，指示語ありの質問を重ねてしたときに生じる破綻である．例を図 2.2 に示す．ここでは，S1 で「スポーツをする目的は何ですか？」と尋ねていて，U1 ではユーザがスポーツをする目的について答えることが予測できる．しかし，S2 では「そのスポーツ」と言って，急に特定のスポーツの話の指示語あり質問 (qs_o_s) が選択されていて，破綻が生じている．

パターン 1 では選択された対話行為内に相応しい発話があるにも関わらず，破綻が生じている．図 2.2 の例でも，S2 で選ばれた特定話題の指示語あり質問 (qs_o_s) の中には,「そのほか健康のために、気を付けていることはありますか？」等相応しい発話が他にある．それにも関わらず，ここでは,「そのスポーツのおすすめポイント～」が選ばれてしまっている．これは，ベースライン手法 2 では行動が対話行為単位であるため，特定話題の指示語あり質問 (qs_o_s) に属する発話集合の中からランダムに発話が選ばれてしまっているためである．このことは，S2 がデフォルト話題の指示語あり質問 (qs_o_d) のときにも同様に言える．

パターン 2

パターン 2 の破綻は，システムが指示語あり質問 (特定話題) もしくは指示語なし質問をしたときのユーザ応答に対して，指示語ありの応答や感謝を返したときに生じる破綻である．例として図 2.3 を示す．ここでは，S1 で「競技は何をご覧になりますか？」と尋ねていて，U1 ではユーザが特定のスポーツを答えることが予測できる．それにも関わらず，S2 では「それは大変ですよね」という，違和感のある指示語あり応答 (re_o_m) を返してしまっている．

パターン 2 では選択された対話行為内に相応しい発話があるにも関わらず，破綻が生じている．図 2.3 の例でも，S2 で選ばれた特定話題の指示語あり応答 (re_o_m)

S1：競技は何をご覧になりますか？(qs_x_s)
U1：野球です
S2(破綻)：それは大変ですよね(re_o_m)
U2：大変ですかね

図 2.3 パターン 2 の破綻例

の中には,「なるほど、面白そうですね。機会があれば私も見てみたいです！」等相応しい発話が他にある．それにも関わらず，ここでは,「それは大変ですよね」が選ばれてしまっている．これは，ベースライン手法 2 では行動が対話行為単位であるため，指示語あり応答 (re_o_m) に属する発話集合の中からランダムに発話が選ばれてしまっているためである．このことは，S1 が特定話題の指示語あり質問 (qs_o_s) であるときや，S2 が感謝 (thank) であるときにも同様に言える．

パターン 3

パターン 3 の破綻は，システムが指示語あり質問 (default) をしたときのユーザ応答に対して，指示語ありの応答や感謝を返した時に生じる破綻である．例として図 2.4 を示す．ここでは，S1 で「おすすめの曲はありますか？」と尋ね，U1 でユーザが曲について答えている．S2 で「具体的に教えてください？」と曲についてさらにさらに尋ね，U2 では曲についてユーザがさらに深く話している．それに対して S3 では「それは大変ですよね。」という違和感のある指示語あり応答 (re_o_m) を返してしまっている．

パターン 3 では，default の指示語あり質問 (qs_o_d) を見ただけでは，次の指示語あり応答 (re_o_m) が破綻しているかどうか分からない．図 2.4 の例で考えると，S2 で何について「具体的に教えてください」と言っているかによって，ユーザの応答は変化し，それに対するシステム応答 S3 の適切さは変わるためである．これを知るためには，S1 を参照し，何について話しているかを知る必要がある．そのため，指示語あり質問（default) の後の指示語あり応答が破綻しているかどうかを知るには，直近の指示語なし質問もしくは指示語あり質問 (特定話題) を参照して，何について話しているかを知る必要がある．

S1：おすすめの曲はありますか？(qs_x_s)
U1：スピッツの楓という曲がおすすめですよ
S2：具体的に教えてください？(qs_o_d)
U2：歌詞が詩的で素敵なんですよね
S3(破綻)：それは大変ですよね。(re_o_m)
U3：大変ですかね

図 2.4 パターン 3 の破綻例

第3章　計算量を考慮した対話行為設計に基づく強化学習

本章では強化学習を用いた発話選択の方策獲得における，計算量を考慮した設計の方針と詳細を述べる．3.1 節では 2 章で述べた強化学習による発話選択手法の問題点を踏まえて，提案手法の大まかな設計方針を述べる．3.2 節では，提案手法の具体的なアプローチである対話行為の詳細化に関して述べる．3.3 節では，詳細化した対話行為を用いた行動と状態の設計に関して詳しく述べる．3.4 節では，破綻削減のために，設計した状態と行動にどのような報酬を設けるかについて論じる．

3.1　方針

対話行為を行動とする手法では，行動選択が対話行為レベルであったために，破綻の生じるシステム発話を選択してしまうことがあった．これを防ぐためには，行動表現を対話行為よりも詳細化し，同一対話行為内の発話でも，どれを選ぶべきなのか順位付けする必要がある．ただし，全ての発話を行動としてしまうと，計算量が増加することが考えられる．

そこで，本研究では，対話行為を行動とする従来手法 [5] で破綻の生じやすかった対話行為内を詳細化して，状態と行動を設計する．これにより，全ての発話を行動とする手法より計算量が少なく，対話行為を行動とする手法より破綻の生じにくい手法を提案する．また，新たに設計した状態と行動に対して，発話レベルでの内容的整合性に関する報酬を設計し，学習することで，破綻の生じるシステム発話の選択を防ぐ．

3.2　対話行為の詳細化

提案手法の行動では，2.4 節で示した，破綻の生じやすい対話行為の順番で，後ろ側に位置する対話行為である以下の 4 つを詳細化したものを用いる．

- 指示語あり質問 (default)
- 指示語あり質問 (特定話題)
- 指示語あり応答 (特定話題+default)
- 指示語あり応答 (特定話題+default)

提案手法の状態では，2.4 節で示した，破綻の生じやすい対話行為の順番で，後ろ側に位置する対話行為である以下の 3 つを詳細化したものを用いる．

- 指示語あり質問 (default)
- 指示語あり質問 (特定話題)
- 指示語なし質問 (特定話題)

3.3　詳細化した対話行為を用いた行動と状態の設計

行動設計では，3.2 節で示した対話行為内を発話ごとに分割して，行動とする．従来の対話行為と，新たに設計した行動 ID の対応を表 3.1 に示す．詳細化すべき対話行為内の発話に，ID をふって行動としている．基本的には 1 行動につき 1 発話としたが，同じような役割を持つ発話に関してはまとめて 1 行動とする．また，これら以外の対話行為内の発話に関しては，対話行為内からランダムに選んでも問題ないものとして，1 つの対話行為を 1 つの行動とする．

対話行為を行動とする従来手法の研究 [5] で用いられた，Hazumi1902[9] から抜粋された 117 個の発話集合を用いると仮定したときの，行動数を示す．詳細化すべき対話行為である (a) 指示語あり質問 (default)，(b) 指示語あり質問 (特定話題)，(d) 指示語あり応答 (特定話題+default，(g) 感謝は，役割がほぼ重複する発話を除いてそれぞれ 7 個，12 個，26 個，2 個の発話があり，これらを行動とすると，合わせて 47 個である．また，これら以外の詳細化しない対話行為は 4 個あるため，行動数は全部で 51 個となる．

状態設計では，3.2 節で示した対話行為内を発話ごとに分割して，状態とする．従来の対話行為と，新たに設計した状態 ID の対応を表 3.2 に示す．詳細化すべき対話行為内の発話に，ID をふって状態としている．また，これに加えて，2.4 節のパターン 3 で述べたように，指示語あり質問 (default) に関しては，直近の指示語あり質問 (特定話題) か指示語なし質問を参照しなければ質問内容が分からない．そのため，

表 3.1 提案手法の行動

対話行為	行動 ID	行動数
(a) 指示語あり質問 (default)	qs_o_d_0 〜 qs_o_d_6	7
(b) 指示語あり質問 (特定話題)	qs_o_s_0 〜 qs_o_s_11	12
(c) 指示語なし質問 (特定話題)	qs_x_s	1
(d) 指示語あり応答 (特定話題+default)	re_o_m_0 〜 re_o_m_25	26
(e) 指示語なし応答 (default)	re_x_d	1
(f) 指示語なし応答 (特定話題+default)	re_x_m	1
(g) 感謝	thank_0 〜 thank_1	2
(h) 情報提供	io	1

表 3.2 中の (j)，(k) のように，この組み合わせを状態として表現する．基本的には 1 状態につき 1 発話としたが，同じような役割を持つ発話に関してはまとめて 1 状態とする．また，これら以外の対話行為内の発話に関しては，1 つの対話行為を 1 つの状態とする．

対話行為を行動とする従来手法の研究 [5] で用いられた，Hazumi1902[9] から抜粋された 117 個の発話集合を用いると仮定したときの，状態数を示す．詳細化すべき対話行為 (b) 指示語あり質問 (特定話題)，(c) 指示語なし質問は役割がほぼ重複する発話を除いてそれぞれ 16 個，37 個の発話がある．また，これに加えて，(j)，(k) に該当する状態はそれぞれ 16 通りと 37 通りある．また，これら以外の詳細化しない対話行為は 7 個ある．そのため，合計の状態数は 113 となる．

3.4 報酬設計

提案手法では，連続する 2 つのシステム発話の整合性に対して報酬を与える．これは，対話行為を行動とする手法では，2.4 節で述べたように，現在のシステム発話から予測されるユーザ発話と，次に選択したシステム発話がかみ合っていないことが主な破綻の原因であったためである．図 2.2 を例にとって考えると，S1 では「スポーツをする目的は何ですか？」と尋ねているため，U1 でユーザはスポーツの目的について答えるものと予想できる．それに対して,「そのスポーツのおすすめポイントを教えてください」という質問は明らかにかみ合わず，破綻を起こすことが予想される．そのため，事前に「スポーツをする目的は何ですか？」という発話の後に「そのスポーツのおすすめポイントを教えてください」という発話が来るのは不自然

表 3.2 提案手法の状態

対話行為	状態 ID	状態数
(a) 指示語あり質問 (default)	qs_o_d	1
(b) 指示語あり質問 (特定話題)	qs_o_s_0 〜 qs_o_s_15	16
(c) 指示語なし質問 (特定話題)	qs_x_s_0 〜 qs_x_s_36	37
(d) 指示語あり応答 (特定話題+default)	re_o_m	1
(e) 指示語なし応答 (default)	re_x_d	1
(f) 指示語なし応答 (特定話題+default)	re_x_m	1
(g) 感謝	thank	1
(h) 情報提供	io	1
(i) 話題変更	ct	1
(j) 指示語あり質問 (特定話題) →指示語あり質問 (default)	qs_o_d_qs_o_s_0 〜 qs_o_d_qs_o_s_15	16
(k) 指示語なし質問 →指示語あり質問 (default)	qs_o_d_qs_x_s_0 〜 qs_o_d_qs_x_s_36	37

であるという関係を表現する．

具体的には，まず各行動に，この状態 (発話) の後に来てはいけないという状態集合 (blacklist) を設ける．行動が選択されたときに，その前の状態 (システム発話) が，その行動の blacklist 内にあれば負の報酬を与える．図 3.1 は，選択された 2 つの発話とその状態，行動に対して，どのように報酬が付けられるのかを表している．この例では「スポーツをする目的〜」という発話の後に,「そのスポーツの〜」という発話が選ばれており，それぞれが状態 qs_x_s_18，行動 qs_o_s_0 として取得されている．ここで，行動 qs_o_s_0 の blacklist を確認し，状態 qs_x_s_18 が含まれていれば，この状態と行動の組み合わせに負の報酬を付ける．行動 qs_o_s_0 の blacklist は表 3.3 のように表される．この表では，行動の列に書かれた特定の行動に対する blacklist(状態集合) が blacklist_1 以降の列に記述されている．この例では，行動 qs_o_s_0 の blacklist に，状態 qs_x_s_18 が付けられている．これは,「スポーツをする目的〜」という発話の後には，ユーザがスポーツの目的に関して答えることが予測でき，その後に「そのスポーツの〜」という特定のスポーツについて尋ねる発話は明らかに噛み合わないためである．これによって，状態 qs_x_s_18 と行動 qs_o_s_0 の組み合わせに負の報酬が与えられ，この順番では選択されにくくなる．

また，直前のシステム発話を見ただけではユーザ発話を予測できず，blacklist を付けられないケースもある．2.4 で述べたパターン 3 の破綻のように，選択した発話 (行動) の直前の発話 (状態) が default の指示語あり質問であるために，ユーザ発話が予測しきれない場合，この状態を blacklist に入れることはできない．そのため，default

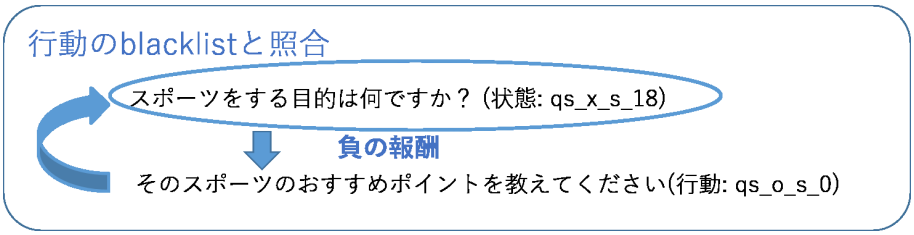

図 3.1 パターン 1,2 に対応する報酬の付け方の例

表 3.3 blacklist の例 (パターン 1,2)

行動	blacklist_1(状態)	blacklist_2(状態)	blacklist_3(状態)
qs_o_s_0	qs_x_s_18	•••	•••

の指示語あり質問のあとの発話 (行動) の blacklist を参照するときには，直近の指示語なし質問か，特定話題の指示語あり質問を状態として取得し，それを blacklist に照合して報酬を決定する．報酬を与える対象としては，まず default の指示語あり質問の状態と，直近の指示語なし質問もしくは特定話題の指示語あり質問の状態を組み合わせた状態と行動の組み合わせである．図 3.2 は，選択された複数の発話とその状態，行動に対して，どのように報酬が付けられるのかを表している．この例では,「具体的に教えてください？」という発話のあとに,「それは大変ですよね。」という発話が選ばれており，それぞれ状態 qs_o_d_2，行動 re_o_m_12 が取得されている．しかし,「具体的に教えてください？」を見ただけではユーザの発話は予測できず,「それは大変ですよね。」の選択が適切かどうかも分からない．そのため,「それは大変ですよね。」という発話の行動である，re_o_m_12 の blacklist を参照する際には，状態として，直近の指示語なし質問である「おすすめの曲〜」という発話の状態 qs_x_s_36 を照合する．行動 re_o_m_12 の blacklist は表 3.4 のように表される．この例では，行動 re_o_m_12 の blacklist に，状態 qs_x_s_36 が付けられている．これによって，qs_x_s_36 と qs_o_d_2 の組み合わせを表現した状態 qs_o_d_qs_x_s_36 と，行動 re_o_m_12 の組み合わせに負の報酬が与えられる．

表 3.4 blacklist の例 (パターン 3)

行動	blacklist_1(状態)	blacklist_2(状態)	blacklist_3(状態)
re_o_m_12	qs_x_s_36	•••	•••

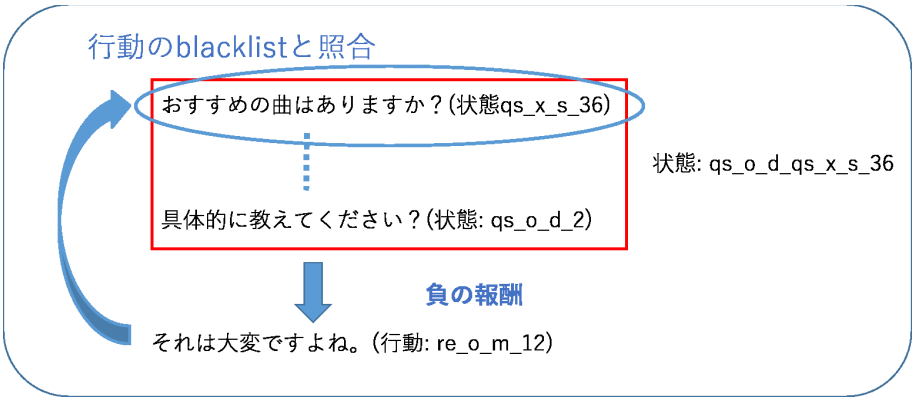

図 3.2 パターン 3 に対応する報酬の付け方の例

第4章　評価実験

本章では提案手法を用いたシステムについて，2つのベースラインと比較し，計算量と破綻の生じにくさの2つの観点から評価を行った．4.1節では全ての手法に共通する実験条件について述べる．4.2節では比較対象のベースライン手法の設計と実装の詳細，比較によって示したい要素について述べる．4.3節では，今回の実験での，提案手法の実装の詳細について述べる．4.4節ではそれぞれの手法を用いて学習したモデルで実際に対話を行った結果を，一定の基準に沿って破綻にアノテートし，破綻の生じにくさを比較した．4.5節ではそれぞれの手法に関して，Qテーブルのサイズに応じたエピソード数での学習時間を比較した．

4.1　共通する実験条件

2つのベースラインシステムと提案手法では，発話集合と学習方法に関して共通したものを用いる．

発話集合として，雑談対話コーパスHazumi1902[9]から抜粋した117個の発話に話題変更の発話を4個加えた121個を扱う．具体的な内容は，話題「スポーツ」「音楽」「食事」「旅行」の4つの話題で用いられる発話に，どの話題にも用いられる「default」の発話を加えたものである．

学習方法は，式2.1によってある状態である行動をとる価値(Q値)を更新していくQ学習を用いた．パラメータ設定は，学習率αを0.1，割引率γを0.9とした．また，システム発話とユーザ発話のセットを1交換と呼び，1エピソードを10交換として学習を行った．エピソードは必ずシステムの話題提供の発話である「(話題)の話をしましょう」から始まる．話題は「スポーツ」「音楽」「食事」「旅行」からランダムで選択され，そのエピソード内では他の話題の発話は選択しない．1エピソード10交換が終了すると，再びシステムの話題提供から対話が始まり，これを定めたエピソード数分繰り返して学習を行う．エピソード数は，ベースライン手法1を基準に，おおよそQテーブルのサイズ(状態数$\times$行動数)に比例するものとした．

4.2　2 つのベースライン手法の設計

2 つのベースライン手法においては，対話の際の学習済み Q テーブルを用いた発話選択において，共通した方法を用いた．具体的には，ある状態での各行動の Q 値を取得し，それらを softmax 関数 (式 4.1) にかけて，出力された確率から行動を確率的に選択する．

以下にベースライン手法それぞれに特有の設計を示す．

$$y_i = \frac{e^{x_i}}{\sum_{k=1}^{n} e^{x_k}} (i = 1, 2, \cdots, n) \tag{4.1}$$

4.2.1　発話集合を対話行為に分類して行動とする手法

一つ目のベースライン手法は，システム発話集合を対話行為に分類して行動とする手法である．

この手法と提案手法を比較することで，状態や行動の詳細化による提案手法の破綻の生じやすさへの影響を検証する．

状態，行動は 2.3 節に示した通りであり，状態数は 24，行動数は 8 であった．状態数 × 行動数は 192 であるため，Q テーブルのサイズは 192 となった．エピソード数は 1000 で学習を行った．

報酬設計は以下の通りである．

1. ユーザ心象値（単一）(+1, 0, −1)
2. ユーザ心象値（連続）(+5, 0, −5)
3. 簡易的なシステム対話行為の連続性 (+0 〜 +10)
4. 特定の名詞を含むユーザ発話に対するシステム行動 (+10, 0, −10)
5. システム行動「thank」の選択 (0, +10)

ユーザ心象値（単一）の報酬では，状態であるユーザの心象が低ければ −1，高ければ+1 の報酬をつける．ユーザ心象値（連続）の報酬では，ユーザの心象が 3 連続で高ければ+5，低ければ −5 の報酬をつける．これらユーザ心象に関する報酬は，システムがユーザの心象がより高くなる行動を選択するために設計されている．

簡易的なシステム対話行為の連続性への報酬は，簡易対話行為の連続性に人手で+1 から+10 の報酬をつける．報酬の付け方を表 4.1 に示す．表の左側がターン $t-1$ の

表 4.1 簡易的なシステム対話行為の連続性への報酬

$t-1\backslash t$	質問	応答	情報提供
質問	3	3	4
応答	5	1	4
情報提供	9	1	0
話題変更	10	0	0

簡易対話行為を表しており，そこからターン t で上側の簡易対話行為が選ばれたときの報酬が示されている．この報酬は，質問をした後に応答を返すような流れは自然であるが，応答を返した後にもう一度応答を返すのは不自然であるというような，簡易対話行為レベルでの自然さを表現している．

特定の名詞を含むユーザ発話に対するシステム行動への報酬では，ユーザ発話に特定名詞（固有名詞等）が含まれる場合に，指示語ありの対話行為を選べば+10，含まれない場合に選べば -10 の報酬をつける．この報酬は指示語が適切なタイミングで使われるように設計されている．

システム行動「thank」の選択への報酬は，話題変更の前に感謝の対話行為が選ばれた場合，+10 の報酬をつける．この報酬は感謝の対話行為は，一つの話題が終わるときに付けるのが適切であるという考えのもと，設計されている．

4.2.2 発話全てを行動とする手法

二つ目のベースライン手法は，システム発話集合中の発話をそのまま状態，行動とするものである．

この手法と提案手法を比較することで，提案手法の計算量削減への有効性と，状態数や行動数削減による破綻の生じやすさへの影響を検証する．

状態と行動は発話集合中の発話全てであるが，話題変更の 4 発話はエピソードの最初に提示して，それ以降は選択しないものとしたため，行動には含めないこととした．状態数が 121，行動数が 117 となるため，Q テーブルのサイズは 14157 であった．ベースライン手法 1 との Q テーブルのサイズの比例関係を考えて，この手法ではエピソード数 76000 で学習を行った．

報酬としては，提案手法と同様に，まず各行動に，この発話の後に来てはいけないという発話集合（blacklist）を設ける．次に，行動が選択されたときに，その前のシステム発話が，その行動の blacklist 内にあれば -20 の報酬を与える．

4.3　計算量を考慮した行動設計に基づく手法の実装

提案手法における行動と報酬の設計は，3.3 節に示した，ベースライン手法 1 の対話行為を一部詳細化したものである．状態数が 113，行動数が 51 であるため，Q テーブルのサイズは 5763 となった．ベースライン手法 1 との Q テーブルのサイズの比例関係を考えて，提案手法ではエピソード数 30000 で学習を行った．

報酬に関しては，3.4 節に示したものであるが，ベースライン手法 2 と同様に，与える負の報酬の値を一律 −20 に設定した．

学習済み Q テーブルを用いた対話での発話選択では，学習した Q テーブルと，ベースライン手法 1 での学習済み Q テーブルの両方を用いる．まず，ベースライン手法 1 の状態を取得し，その学習済み Q テーブルを用いて，対話行為を選択する．次に，本章で述べた提案手法での状態を取得し，その状態での各行動の Q 値を取得する．Q 値を取得した行動の中で，選択した対話行為に属するもののみを取り出す．取り出された行動の Q 値を正規化し，softmax 関数 (式 4.1) にかけることで，その行動の選択確率を決定し，行動選択を行う．

発話選択の全体像の例は図 4.1 のようになる．まず,「最近はまっている食べ物はありますか?」というシステム発話と，それに対する「バナナにはまっています」というユーザ発話，入力された心象の組み合わせから状態が取得される．その状態をもとに，対話行為を行動とする手法によって re_o_m という対話行為が選択されたとする．一方，提案手法側では,「最近はまっている食べ物はありますか？」という発話を状態として，その状態での行動（発話）全ての Q 値が取得される．例では，これらの行動 (発話) の中で，re_o_m に属しており，Q 値の高い「とても美味しそうですね」が確率的に選択されている．

4.4　破綻の生じにくさに関する評価

本節では提案手法の設計が破綻削減に有効であることを示す．そのため，提案手法及びベースライン手法 2 つの学習済み Q テーブルを用いてテキスト対話を行い，生じた破綻の数を比較して，破綻の生じやすさの評価を行った．

テキスト対話は，システム発話とユーザ発話の組み合わせを 1 交換として，10 交換 1 セットで行った．話題「スポーツ」「音楽」「食事」「旅行」の４つに関して 5 セットずつ計 20 セット 200 交換を評価の対象とした．特定話題に関する対話では，その特定話題の発話と default 発話が全体の発話集合となり，発話はその中から選択される．

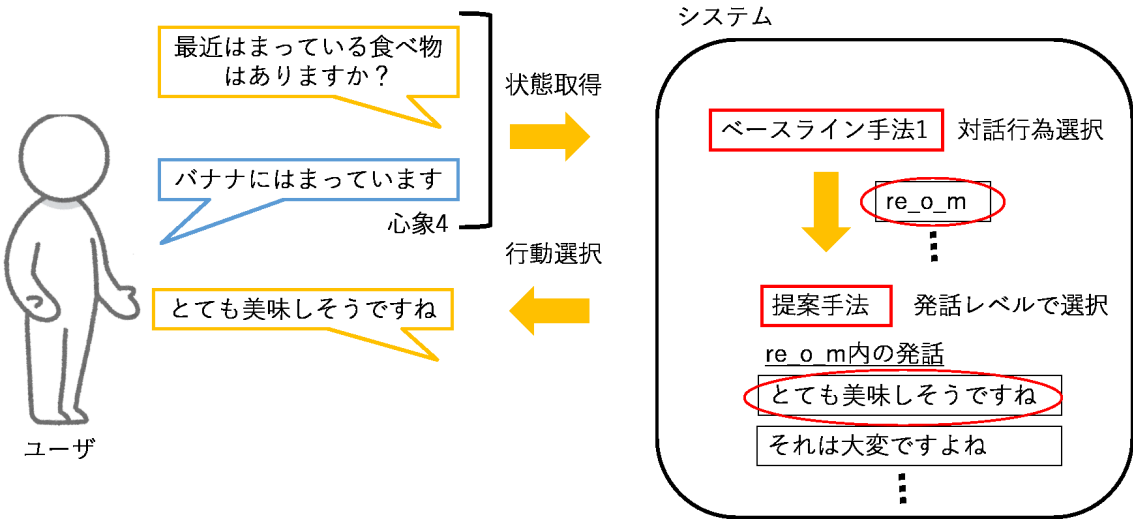

図 4.1 発話選択方法

評価は東中らの研究 [10] で用いられた破綻の基準をもとに行った．以下の基準をもとに各交換に対して○△×でアノテーションをつけた．

- ○破綻ではない:
  当該システム発話のあと対話を問題無く継続できる．
- △破綻と言い切れないが，違和感を感じる発話:
  当該システム発話のあと対話をスムーズに継続することが困難．
- ×あきらかにおかしいと思う発話．破綻:
  当該システム発話のあと対話を継続することが困難．

△や×の数と，全交換数に対する△や×の割合によって破綻の生じやすさの評価を行った．また，東中らの研究 [10] によると，これらのアノテーション基準では，△と×の付け方が個人によってばらつきが大きい．そのため，△と×の和も算出し，同様に評価を行った．

ベースライン手法と提案手法の比較を表 4.2 と 4.3 に基づいて説明する．表 4.2 を見ると，ベースライン手法 1 では△の数が 21 個，×の数が 13 個であるのに対して，提案手法では△の数が 4 個，×の数が 6 個と減少している．また，それぞれの全交換数に対する割合も△では 11%から 2%，×では 6.5%から 3%とかなり減少している．また，4.3 を見ると，提案手法では△と×の数が 24 個減っていて．割合も 17%から 5%となっていて，全体的にも違和感のある発話はかなり減っている．

表 4.2 ○と△と×による破綻の生じやすさの比較

	○	△	×	△の割合 (%)	×の割合 (%)
提案手法	190	4	6	2%	3%
ベースライン手法 1	166	21	13	11%	6.5
ベースライン手法 2	187	8	5	4%	2%

表 4.3 ○と△×による破綻の生じやすさの比較

	○	△と×	△と×の割合 (%)
提案手法	190	10	5.0%
ベースライン手法 1	166	34	17%
ベースライン手法 2	187	13	6.5%

また，ベースライン手法 2 と提案手法の比較を表 4.2 と表 4.3 に基づいて説明する．表 4.2 を見ると，ベースライン手法 2 では△の数が 8 個，×の数が 5 個であるのに対して，提案手法では△の数が 4 個，×の数が 6 個となっていて，△の数は多少減っているが，×の数は多少増えている．また，それぞれの全交換数に対する割合も同様である．また，4.3 を見ると，提案手法では△と×の数が 3 個減っていて，割合も 6.5%から 5%となっている．これらはほぼ同程度であるとみなせる．

これらのことから，提案手法の設計が破綻の削減に有効であることが分かった．

以下で提案手法での，破綻改善への設計の有効性について述べる．2.4 節での，パターン 1 から 3 の破綻について，ベースライン手法 1 で生じた破綻例と，提案手法での改善例を示し，提案手法の設計との関連を述べる．

パターン 1 の破綻改善

ベースライン手法 1 では，パターン 1 の破綻と同様の破綻が生じた．図 4.2 を見ると，指示語なし質問 (qs_x_s) に対して，不適切な指示語あり質問 (qs_o_s) が選択されたことで破綻が生じている．この例では，S1 で旅行の目的を聞いており，U1 でユーザが旅行の目的について答えることが予測できる．それにも関わらず，S2 で「その旅先」と具体的な旅先について尋ねているため，破綻が生じている．

一方，提案手法では，パターン 1 の破綻に関する改善が見られた．同じ例ではないが，図 4.3 を見ると，指示語なし質問 (qs_x_s) に対して，適切な指示語あり質問 (qs_o_s) を選択することができている．この例では，S1 で好きなスポーツについて

S1: 旅行に行くときはなにか目的はありますか？(qs_x_s)
U1: 美しい景色を見ることですかね
S2(破綻): その旅先で何かハプニングはありましたか!(qs_o_s)
U2: どの？

図 4.2 ベースライン手法 1 でのパターン 1 の破綻例

尋ね，ユーザは何らかのスポーツを答えることが予測される．S2 では特定のスポーツをテレビで観戦するかを尋ねており，これは破綻の生じない発話となっている．

このような破綻の生じない発話を選択できた理由として，提案手法での，対話行為の詳細化と，システム発話の内容的整合性への報酬が有効に働いていることが挙げられる．S2 で選択された「そうですか。テレビでの観戦はされるんですか?」という発話は，指示語あり質問 (特定話題) が詳細化された，qs_o_s_0 という行動である．また，S1 での「好きなスポーツはなんですか」という発話は，指示語なし質問が詳細化された，qs_x_s_20 という状態として取得されている．報酬設計では，qs_o_s_0 という行動の blacklist には qs_x_s_20 という状態を入れていなかったが，他の不適切な指示語なし質問の発話（状態）は blacklist に入れていた．これにより，相対的に「そうですか。テレビでの観戦はされるんですか？」という，破綻の生じない発話が選ばれやすくなるように学習されたと考えられる．

S1: 好きなスポーツはなんですか？(状態: qs_x_s_20)
U1: 野球です
S2: そうですか。テレビでの観戦はなされるんですか？(行動: qs_o_s_0)
U2: ええ主にテレビで見ます

図 4.3 提案手法でのパターン 1 の破綻改善

パターン 2 の破綻改善

ベースライン手法 1 では，パターン 2 の破綻と同様の破綻が生じた．図 4.4 を見ると，指示語なし質問 (qs_x_s) に対して，不適切な指示語あり応答 (re_o_m) が選択されたことで破綻が生じている．この例では，S1 でスイーツを買う場所について聞

S1：ちょっと特別なスイーツを買うときは近くのお店とか
　　「デパートの地下」とか、利用しますか？(qs_x_s)
U1：やはりコンビニですね
S2(破綻)：**面白いですよね** (re_o_m)
U2：面白いかな

図 4.4 ベースライン手法 1 でのパターン 2 の破綻例

S1：ちょっと特別なスイーツを買うときは近くのお店とか
　　「デパートの地下」とか、利用しますか？(状態：qs_x_s_24)
U1：やはりコンビニですね
S2：　**いいですね，とても興味深いです！**(行動：re_o_m_11)
U2：そうですか

図 4.5 提案手法でのパターン 2 の破綻改善

いており，U1 でユーザがその場所について答えることが予測できる．それにも関わらず，S2 で「面白い」という反応を返しているため，違和感を感じる発話になっている．

一方，提案手法では，パターン 2 の破綻に関する改善が見られた．ベースライン手法 1 と同様の例で，図 4.5 を見ると，指示語なし質問 (qs_x_s) に対して，適切な指示語あり応答を返すことができている．この例では，同様の質問に対して,「いいですね，とても興味深いです！」という破綻の生じない発話を返している．

このような破綻の生じない発話を選択できた理由として，提案手法での，対話行為の詳細化と，システム発話の内容的整合性への報酬が有効に働いていることが挙げられる．S2 で選択された「いいですね，とても興味深いです！」という発話は，指示語あり応答が詳細化された，re_o_m_11 という行動である．また，S1 での「ちょっと特別なスイーツを買うときは〜」という発話は，指示語なし質問が詳細化された，qs_x_s_24 という状態として取得されている．報酬設計では，re_o_m_11 という行動の blacklist には qs_x_s_24 という状態を入れていなかったが，他の不適切な指示語なし質問の発話 (状態) は blacklist に入れていた．これにより，相対的に「いいですね．とても興味深いです！」という，破綻の生じない発話が選ばれやすくなるように学習されたと考えられる．

パターン 3 の破綻改善

ベースライン手法 1 では，今回の実験でも，パターン 3 の破綻と同様の破綻が生じた．図 4.4 を見ると，default の指示語あり質問 (qs_o_d) に対して，不適切な指示語あり応答 (re_o_m) が選択されたことで破綻が生じている．この例では，S1 で甘いものと辛いもののどっちが好きかを尋ねていて，続いて S2 でその具体例について尋ねる形になっている．ここではユーザが甘いものか辛いものかの具体例を答えることが予測されるが，S3 で「すごい」という反応を返しているため，破綻が生じている．

一方，提案手法では，パターン 3 の破綻に関する改善が見られた．同じ例ではないが，図 4.4 を見ると，default の指示語あり質問 (qs_o_d) に対して，適切な指示語あり応答を返すことができている．この例では，S1 でカラオケで歌う曲について尋ね，続いてその具体例について尋ねる形になっている．S3 では「そうなんですか〜」という破綻の生じない発話を返している．

このような破綻の生じない発話を選択できた理由として，提案手法での，対話行為の詳細化と，システム発話の内容的整合性への報酬が有効に働いていることが挙げられる．S3 で選択された「そうなんですね。新しい知識が〜」という発話は，指示語あり応答が詳細化された，re_o_m_17 という行動である．また，S1 で選択された「カラオケでよくなにをうたうのですか？」という発話と，S2 で選択された「具体的にはどういったものですか」という発話の組み合わせは，qs_o_d_qs_x_s_16 という状態として取得されている．報酬設計では，re_o_m_17 という行動の blacklist には qs_o_d_qs_x_s_16 という状態を入れていなかったが，他の不適切な指示語なし質問と指示語あり質問 (default) の組み合わせの状態は blacklist に入れていた．これにより，相対的に「そうなんですか〜」という破綻の生じない発話が選ばれやすくなるように学習されたと考えられる．

4.5 計算量に関する評価

本節では，提案手法の設計が計算時間の削減に有効であることを示す．本実験では，提案手法，ベースライン手法 1，ベースライン手法 2 に関して学習を行った際の計算時間を比較した．それぞれの Q テーブルのサイズ，エピソード数，学習にかかる総計算時間，1 エピソード当たりの計算時間を表 4.4 に示す．これらを用いて Q テーブルのサイズによるエピソード数の変化，また，エピソード数の変化による計算時間の変化，手法の設計が計算時間に与える影響等について考察する．また，4.4

S1：甘いものと、からいものどちらがお好きですか？(qs_x_s)
U1：甘い物です
S2：具体的には教えてもらえますか？(qs_o_d)
U2：チョコとかですかね
S3(破綻)：すごい！(re_o_m)
U3：すごいかな

図 4.6 ベースライン手法 1 でのパターン 3 の破綻例

S1：カラオケでよくなにをうたうのですか？(状態：qs_x_s_16)
U1：やはりスピッツですね
S2：具体的にはどういったものですか？(状態：qs_o_d_qs_x_s_16)
U2：チェリーとかロビンソンとかです
S3：そうなんですね。新しい知識が増えました。
私も一度聞いてみたいと思います！(行動：re_o_m_17)
U3：ええぜひ

図 4.7 提案手法でのパターン 3 の破綻改善

節で示した破綻の生じにくさと総計算時間両方について考慮することで，手法の総合的な性能を述べる．提案手法に関しては，破綻の生じにくさの実験において，提案手法とベースライン手法 1 両方の学習済み Q テーブルを用いて発話選択を行った．そのため，提案手法の性能に関しては，提案手法とベースライン手法 1 両方の学習の成果であると言える．ゆえに，表 4.4 の提案手法の Q テーブルのサイズやエピソード数，計算時間には，ベースライン手法 1 の値を足し合わせたものを記述している．

提案手法とベースライン手法 1 の比較を表 4.4 に基づいて説明する．まず，提案手法とベースライン手法 1 の 1 エピソード当たりの計算時間を比べると，ベースライン手法 1 が提案手法の約 1.4 倍程度になった．これは，提案手法に比べて，ベースライン手法 1 ではプログラム上で最も時間のかかる報酬を与える部分で，与える報酬の種類が多く，参照する状態や行動が多いことに起因する．また，提案手法ではベースライン手法 1(対話行為を行動とする手法) の対話行為を詳細化して状態と行動を設計している．そのため，Q テーブルに関しては，提案手法がベースライン手法 1 の約 31 倍になっている．今回の実験では，学習するエピソード数を Q テーブル

にサイズに比例するものとしたため，提案手法では，エピソード数が，ベースライン手法 1 の約 31 倍になっている．これらのことから，総計算時間に関して，提案手法がベースライン手法 1 の約 21 倍となっている．

提案手法とベースライン手法 1 の，計算時間と破綻の生じにくさの関係について述べる．本実験と同じ条件で学習された Q テーブルを用いて対話を行った，4.4 節の破綻の生じにくさに関する評価実験では，提案手法とベースライン手法 1 では，提案手法手法が優れていることが示された．また，本実験では，提案手法の総計算時間はベースライン手法 1 の約 21 倍であった．これらのことから，提案手法は，計算時間は増えるが，破綻の生じにくさに関して，ベースライン手法 1 よりも優れた性能を持つ手法といえる．

提案手法とベースライン手法 2 の比較を表 4.4 に基づいて説明する．まず，提案手法とベースライン手法 2 の 1 エピソード当たりの計算時間を比べると，提案手法がやや多いものの，ほぼ同じとみなせる程度であった．これは，提案手法とベースライン手法 2 では，プログラム上で最も時間のかかる報酬を与える部分の設計がほぼ同じであることが原因であると考えられる．また，提案手法ではベースライン手法 1(対話行為を行動とする手法) で破綻の生じやすい対話行為のみを詳細化して状態と行動の設計に用いている．一方，ベースライン手法 2 では，全ての発話を状態と行動の設計に用いている．これによってベースラインライン手法 2 では Q テーブルのサイズが大きく，提案手法の約 2.4 倍になっており，エピソード数も約 2.4 倍になっている．1 エピソード当たりの計算時間があまり変わらないことと，学習するエピソード数がベースライン手法 2 では約 2.4 倍であることから，総計算時間に関してもベースライン手法は約 2.2 倍程度になっている．

提案手法とベースライン手法 2 の計算時間と破綻の生じにくさの関係について述べ，提案手法が計算時間の削減に有効であることを示す．本実験と同じ条件で学習された Q テーブルを用いて対話を行った，4.4 節の破綻の生じにくさに関する評価実験では，提案手法とベースライン手法 2 は破綻の生じにくさに関して，同程度の性能を持つことが示された．また，本実験では，ベースライン手法の総計算時間は提案手法の約 2.2 倍であった．これらのことから，提案手法では，約 2.2 倍の計算時間がかかるベースライン手法 2 と比べて，破綻の生じにくさに関して同程度の性能を実現できているといえる．よって，提案手法の設計は計算時間の削減に有効であることが示された．

表 4.4 計算時間の比較

	Q テーブルのサイズ	エピソード数	総計算時間	計算時間 (1 エピソード)
提案手法	5955	31000	2777 秒	0.08958 秒
ベースライン手法 1	192	1000	130 秒	0.130 秒
ベースライン手法 2	14157	76000	6118 秒	0.08050 秒

第5章　結論

本研究では，強化学習を用いた傾聴型対話システムの発話選択手法に関して，計算量を考慮した行動設計を行い，破綻の減少を目指した．発話を対話行為に分類して状態や行動を設計する手法を分析し，破綻の生じやすい対話行為の順番を明らかにした．提案手法では，それらの対話行為内のみを詳細化して状態や行動を設計し，連続するシステム発話の内容的整合性に基づいた報酬を設定した．

破綻の生じにくさと計算時間の2つの観点から提案手法の評価実験を行った．実験の結果，提案手法は，破綻の生じにくさの点で，対話行為を行動とする手法よりも優れており．発話全部を行動とする手法と同程度の性能であった．また，提案手法は，計算時間の点で発話全部を行動とする手法よりも優れていることが示された．

本研究では，ユーザの，話を聞いて欲しいという欲求を満たすために，ユーザが気持ちよく対話を続けられるよう破綻の少ない対話システムを目指した．そのために，システム発話から次のユーザ発話を予測し，それに対して破綻を起こすシステム発話は選択されにくくなるよう学習を行った．一方，ユーザ発話を予測しきれないシステム発話に関しては，一部を除いて，次にどのシステム発話が来ても破綻が起こると決め，一律に選択されにくくなるように学習した．これにより，ユーザ発話の内容によっては破綻が起こらないにも関わらず，選択されにくいシステム発話があった．この設計では，本来あり得たシステム発話が選択されづらくなることで，発話候補が減る．これによって，長い間，もしくは複数回対話を行うと，同じような順番での発話選択が多くなり，ユーザに飽きられやすくなってしまう．しかし，ユーザの，話を聞いてほしいという欲求を満たすためには，ある程度長い間，もしくは複数回対話を行うことも重要である．

今後の課題としては，ユーザと長い間，もしくは複数回対話を行っても飽きられないシステムの開発が挙げられる．これの実現方法として，ユーザの応答からも状態を取得することが挙げられる．ユーザ応答から状態を取得することで，ユーザの応答が予測できない発話も，ユーザ状態によって選択してよいかどうかを学習できる．これにより，発話の選択肢が増え，より飽きられないシステムを実現できるのではないかと考えている．

参考文献

[1] 猿渡 洋, 川波 弘道, 鹿野 清宏, “実環境向け音声対話ロボット「キタちゃん」の開発,” **日本ロボット学会誌**, vol. 28, no. 1, pp. 31–34, 2010.

[2] Toyomi Meguro, Yasuhiro Minami, Ryuichiro Higashinaka, and Kohji Dohsaka, “Learning to control listening-oriented dialogue using partially observable Markov decision processes,” *ACM Trans. Speech Lang. Process.*, vol. 10, no. 4, Jan. 2014.

[3] 石田真也, 井上昂治, 中村静, 高梨克也, 河原達也, “傾聴対話システムのための発話を促す聞き手応答の生成,” *SIG-SLUD*, vol. B5, no. 01, pp. 1–6, aug 2016.

[4] 東中竜一郎, 稲葉通将, 水上雅博, *python* **でつくる対話システム**, オーム社, 2020.

[5] 西本遥人, “雑談対話システムにおける対話行為セットの設計と強化学習を用いた発話選択,” **大阪大学工学研究科修士論文（未刊行）**, 2019.

[6] 江頭勇佑, 柴田知秀, 黒橋禎夫, “雑談対話システムにおける強化学習を用いた応答生成モジュールの選択,” **言語処理学会第** *18* **回年次大会論文集**, pp. 654–657, 2012.

[7] 佐藤真, 高木友博, “深層強化学習を用いたシチュエーション対話向け応答選択モデル,” *SIG-SLUD*, vol. B5, no. 02, pp. 116–121, nov 2020.

[8] Zhou Yu, Ziyu Xu, Alan W Black, and Alexander Rudnicky, “Strategy and policy learning for non-task-oriented conversational systems,” in *Proceedings of the 17th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, Los Angeles, Sept. 2016, pp. 404–412, Association for Computational Linguistics.

[9] 駒谷 和範, 岡田 将吾, “複数の主観評定を付与した人システム間マルチモーダル対話データの収集と分析,” **信学技報**, vol. 119, no. 179, pp. 21–26, aug 2019.

[10] 東中竜一郎, 船越孝太郎, “Project Next NLP 対話タスクにおける雑談対話データの収集と対話破綻アノテーション,” *SIG-SLUD*, vol. B4, no. 02, pp. 45–50, 2014.

謝辞

本研究の遂行に際し，多大なる御指導，御鞭撻を賜りました大阪大学産業科学研究所 駒谷和範教授，武田龍准教授に，深く感謝いたします．また，本研究を進めるにあたり，多くの助言を頂いた，大阪大学大学院工学研究科電気電子情報通信工学専攻 奥野尚己氏をはじめとする大阪大学産業科学研究所 駒谷研究室の学生一同に深く感謝いたします．また，日頃よりお世話になりました駒谷研究室技術補佐員 谷端紀久子氏，事務補佐員 松下美佐氏に深く感謝致します．