

ライフログに応じて発話を変えることで ユーザに親密さを感じさせる対話システムの検討

前菌 そよぎ¹ 原直¹ 阿部 匡伸¹

概要: 人とシステムの会話において、ユーザが対話を楽しむためには、システムに対してユーザが親密さを感じるほうが望ましいと考えられる。人とシステムの会話において、会話を重ねるたびにシステムがユーザに関する情報を用いて応答内容を変えると、ユーザはシステムに親密さを感じるのではないかと考える。なお、ユーザに関する情報はライフログから得られると想定する。本稿では、前段階としてユーザのライフログを利用する対話システムを作成し、システムの評価と今後の課題を検討した。主観評価実験の結果、ユーザのライフログから推定した単語の推定結果が正解の際には、ユーザの感情極性(ユーザがポジティブな話題ととらえるか否か)ごとに発話内容を変える必要があることが分かった。また、推定結果が不正解の際には、誤ったスロット種別を用いた応答など、推定誤りによって一貫性がないように感じられることが分かった。

1. はじめに

身近に存在する対話システムやロボットの例として、りんな [1] や, Pepper [2] がある。人とコミュニケーションをとるロボットでは、ロボットと人との間で、自然なインタラクションを形成する必要があると考えられており、そのために、人間同士のコミュニケーションを模倣するアプローチが提案されていたり [3]、対話システムの機能として重要な項目を挙げるために、人と人との対話が分析されている [4][5][6]。

人と人との会話では、自己開示をすることで互いに親密になるということが分かっている [5]。さらに、一般的には親密度の低い人同士の会話よりも、親密度の高い人同士の会話の方が自己開示することが分かっている [6]。また、人とシステムの対話においても、ユーザに親しみやすさを感じさせるために、ユーザの情報を考慮した発話を行うことは効果的な手法であることが分かっている [7][8]。つまり、自己開示をすることでより親密度が高まり、さらに自己開示を促進するという循環が生まれると考えられる。例えば、自己開示することで、互いがどんな時にどんな感情になるかの共通理解がなされ、お互いに気遣うことでさらに親密度が高くなると考える。

本報告では、ユーザの自己開示をライフログとその時の感情を利用することで実現し、人とシステムの間での親密度向上の循環を促す対話方式を検討する。人とシステムが

会話を重ねるたびに、システムがユーザのライフログと感情を蓄積して、それらをもとに応答内容を変えることにより、システムに親密さを感じさせることを目指す。また、使い続けることで、ユーザの体験等、生活の記録を蓄積できるライフログとしての活用も期待できる。

本稿は以下の通りの構成である。第2章では、提案方式を述べる。第3章では、対話システムのプロトタイプについて述べる。第4章では、主観評価実験について述べる。最後に第5で本論文のまとめについて述べる。

2. 提案方式

本報告では、ユーザとの対話からユーザの日々のエピソードを聞き出し、ライフログとして収集する。収集したライフログを基に、対話の応答内容を変えることで、ユーザに対し、親密さを感じさせ、もっと話し続けたいと思わせるような対話システムとなることを目指す。

2.1 音声対話システムの概要

音声対話システムの概要を図1に示す。

まず、図1(a)(緑枠)は、ユーザ発話から発話理解として、ユーザの行った場所やそこでの出来事、ユーザの感情等のエピソードを抽出する。抽出されたエピソードを、ユーザのライフログとしてライフログデータベースに保存する。

続いて、図1(b)(赤枠)は、抽出されたエピソードとライフログデータベースから、応答文候補を生成する。応答文候補の生成部分では、ユーザのエピソードを聞き出す質問

¹ 岡山大学 大学院ヘルスシステム統合科学研究科

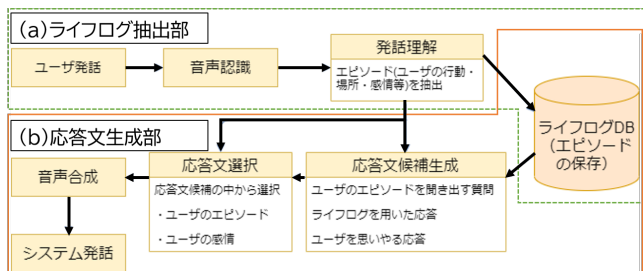


図 1: 音声対話システムの概要

表 1: エピソード種別

エピソード種別	意味
daily_store	日常の買い物
non_daily_store	病院, 美容院
food	飲食店
play	遊び
school	学校
work	仕事
home	家

や、ライフログを用いた応答、ユーザーを思いやる応答などの応答文を生成する。応答文選択部分で、先ほど生成された候補文から、ユーザーの発話内容や、ユーザーの感情を考慮して、最も良いものを選択する。選択された応答文をシステム発話として発話する。

2.2 ライフログ収集の流れ

ユーザーとの会話からライフログをどのように収集するかについて表 1～表 4 を用いて説明する。

システムとユーザーの対話は、エピソード単位で行うものとする。エピソードは、表 1 に示す種別があり、複数のスロットを持つフレーム構造で表現されている。スロットは表 2 に示すようにスロット種別と属性値を持つ。対話の中から、表 3 のようなスロット属性値を抽出する。スロットの属性値は最初、空になっており、空の属性値を埋めるような質問を生成する。例えば、表 3 の Episode3 の対話をする際、システムは空欄である person や traffic の属性値を埋めるような質問を行うことでエピソードを聞き出す。

エピソード種別 (表 1) とエピソードの詳細 (表 3) を日ごとに収集し、表 4 のように蓄積したものをライフログと呼ぶ。なお、一部のスロット属性値が欠損したエピソードも許容する。

2.3 ライフログを用いた対話

このシステムが目指すユーザーとの対話例について 2.3.1 項から 2.3.3 項に示す。システムが目指す応答として、収集されたライフログをもとに、ユーザーの感情極性を推定し、ユーザーの感情極性によって発話が変わる応答がある。ユーザーの感情極性を考慮した応答文として、例えば、ユーザーがある場所に行くときはよくポジティブな感情になっている

表 2: 抽出するエピソードのフレーム表現

スロット種別	属性値例
place	場所名
person	誰と過ごしたか
emotion	感情と感情極性
traffic	交通手段
event	出来事
kind	食べ物・買い物の種類
price	値段
date	期間

表 3: Episode Detail

エピソード ID	place	person	emotion	traffic	event	...
Episode1	スーパー	1人	普通 (even)	徒歩	食材を購入	...
Episode2	バイト	後輩	しんどい (negative)	車	清掃	...
Episode3	イオン		美味しい (positive)		ラーメン	...

表 4: lifelogDB

日時	場所	エピソード種別	エピソード ID
2022/01/01 10:00	岡山市	daily_store	Episode1
2022/01/02 17:00	岡山市	non_daily_store	Episode2
2022/01/03 19:30	倉敷市	food	Episode3

ことがライフログデータベースから分かったとすると、システムは積極的に質問し、別の場所に行くときはよくネガティブになっていることがライフログデータベースから分かったとすると、システムはユーザーを労わる応答をする。

2.3.1 初期の対話例

初期の対話では、基本的に質問のみを行う。システムは、質問によって得られたユーザーのエピソードをライフログとして貯蓄する。

システム こんにちは、今日はどこへ行ったの？

ユーザー 今日はラーメン屋に行ったよ。

システム 誰と行ったの？

ユーザー 友達と行ってきたよ。

システム おいしかった？

2.3.2 一定期間が経過したころの対話例

一定期間が経過すると、システムはユーザーのライフログをある程度収集している。よって、対話システムは、ユーザーのエピソードを聞き出す質問に加えて、ライフログデータベースを用いて応答をする。

システム こんにちは、今日はどこへ出かけたの？

ユーザー 今日はラーメン屋に行ったよ。

システム 先週もラーメン屋に行ってたね。

ユーザー そう、同じラーメン屋さんに行ってきたよ。

システム おいしかった？

2.3.3 さらに期間が経過したころの対話例

さらに一定期間が経過すると、システムは、ユーザーのエピソードを聞き出す質問、蓄積されたライフログを用いた応答に加えて、ユーザーの感情極性をライフログデータベー

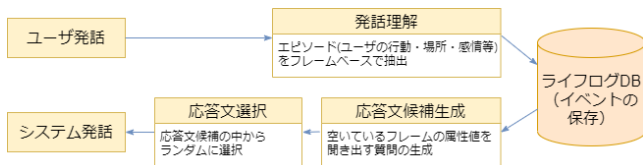


図 2: 対話システムのプロトタイプ

スをもとに予測し、感情極性を考慮した応答をする。次の対話例はユーザがよくネガティブになる場所だとシステムが予測した際の応答である。

システム こんにちは、今日はどこへ行ったの？
ユーザ さっきバイトから帰ってきたよ。
システム お疲れ様。今日のバイトも大変だった？
ユーザ うん、客がたくさん来てとても疲れた。
システム 大変だったね。ゆっくり休んでね。

3. 対話システムのプロトタイプ

本報告で開発した対話システムのプロトタイプを図 2 に示す。また、本稿では対話制御(発話理解, 応答文候補生成, 応答文選択)に関する検討をするため、音声認識や音声合成は行わず、テキストベースの対話システムを実装した。この対話システムは、telegram[9] というアプリケーション上で対話する。まず、システム発話が表示され、そのシステム発話に応じた文がユーザから入力される。この入力された文から、発話理解として、ユーザのエピソードを抽出し、ライフログとして保存する。応答文候補生成では、ユーザのエピソードを聞き出す質問をフレームベースで生成し、生成された応答文候補の中からランダムに応答文を選択する。

3.1 発話理解

発話理解ではユーザの入力文を 7 種類のエピソード種別(表 1)に分類する。また、ユーザの入力文中の単語からユーザの行った場所や出来事、感情などのスロット種別に対応する属性値(表 2)を抽出する。

例えば、ユーザが、「今日は友達ととんかつを食べに行ったよ」という文を入力したとする。この入力文からエピソード種別を推定して、エピソード種別は food であると分類する。

また、ユーザの入力文中の単語から 8 種類のスロット種別(表 2)に対応する属性値を抽出する。例えば、ユーザが「今日は友達ととんかつを食べに行ったよ」という文を入力したとする。スロットの属性値抽出で、システムはユーザの発話内の単語から、person に「友達」という属性値、kind に「とんかつ」という属性値を抽出してスロットの属性値を埋める。

エピソード種別分類には、SVM を用いて、スロットの属性値抽出には CRF を用いて分類を行った [10]。また、学

習データは事前に準備した単語データを組み合わせて機械的に生成された大量の文である。

3.1.1 学習データ

学習データは以下の手順で作成した。

(1) エピソード種別ごとにテンプレートの文を用意

7 種類のエピソード種別ごとに、そのエピソード種別に適したテンプレートの文を 25 文ずつ用意する。この文は「○○へ行ったよ」や「△△を食べたよ」のような文になっており、○○や△△の部分に単語が入る。

(2) スロット種別ごとに単語ファイルを 20 種類用意

(1) のテンプレート文と組み合わせる単語を用意する。単語はスロットの属性値に入る可能性のある単語をスロット種別ごとに単語ファイルに分けて用意する。また、各単語ファイルの単語数は 50~800 単語である。単語数の少ないファイルはエピソード種別 school の place のスロット種別で学校内の施設の場所などである。逆に単語数の多いファイルはエピソード種別 food の place のスロット種別で飲食店の固有名詞を挙げている。なお、place のスロット種別に関しては、岡山県内の場所に限定して単語を用意した。emotion, traffic, person のスロット種別は各エピソード種別ごとに共通だが、それ以外のスロット種別はエピソード種別ごとに別々のファイルが用意される。例として、エピソード種別の daily_store のスロット種別 place に入る単語を集めた単語ファイルを以下に示す。daily_store は日常の買い物というエピソード種別な為、place のスロット属性値には、スーパーや薬局、本屋などの日常の買い物に使われる店名などの単語が単語ファイルに収集されている。ファイル中の単語はホットペッパーの店名サーチ API*1 や岡山市オープンデータ*2 を基に手作業で作成した。

daily_store.place の単語ファイル

```

イオン
100 円ショップ
bookstore
...
  
```

(3) テンプレートの文に単語ファイルを挿入

単語ファイルに入っている単語からランダムに選択されたものをテンプレート文と組み合わせ、その文を 1000 倍にするスクリプトを実装し、学習データの文を作成する。よって、1 エピソード種別あたり、25 万文の学習データが作成される。

(1)~(3) に追加で、今回の実験に協力してもらったユーザの行動に関連がある場所名などのデータも学習データとして準備する。ユーザの行動に関連があるデータがないと

*1 <http://webservice.recruit.co.jp/hotpepper/shop/v1/>

*2 <https://okayama-city.dataeye.jp/>

表 5: IOB タグ付けの例

友達	と	岡山	駅	に	行っ	た
B-friend	O	B-place	I-place	O	O	O

表 6: 相槌一覧

へえ!	そうなんだ。	うんうん!	なるほど	へー!
ふむふむ。	ええ!	ほうほう!	いいね。	ほお!
そうなのか!	おー!	おお!		

飲食店の名前等の固有名詞が出てきた際にエピソード種別やスロット種別の分類が上手くいかない。よって、ユーザの行動に関連があるデータを単語ファイルに追加し、学習データを作成する。ユーザの行動の収集方法は 4.1 項に示す。

3.1.2 エピソード種別の分類

学習データを SVM を用いてエピソード種別 7 種類に分類する。SVM(Support Vector Machine) とはデータを 2 つのクラスに分離する超平面のうち、各データから最も離れているものを決定する手法であり、主に二値分類に用いられている。二値分類の SVM を用いてマルチタスク分類 (7 種類の分類) をする方法について説明する。まず、学習時には 7 つあるラベルから 2 つ選び、一方を正例、もう一方を負例として学習し、21 個の分類期を作成する。推論時には、21 個の分類器の結果を多数決する。多数決で最も多かったエピソード種別を推定結果とする。

3.1.3 スロット種別に対応する属性値の抽出

学習データを CRF を用いて系列ラベリングする。系列ラベリングとは、入力にデータ列を受け取り、出力として個々のデータにラベルを付与するものである。CRF とは、系列ラベリングに対して、各データに各ラベルが割り当てられる確率を得ることができる手法である。学習データのある文から、文を形態素解析し、形態素、品詞、ラベルを取得する。ラベルは IOB 形式で、各属性の単語列の最初の単語に B を、最初以外の単語に I を、属性と関係ない単語に O をつける。例えば、「友達と岡山駅に行った」の場合、表 5 のようなスロット種別が付与される。

CRF を用いて系列ラベリングを行うには、入力単語列を特徴ベクトルに変換する必要がある。特徴量として単語、品詞、直前の単語や品詞、自身が文末かどうかという情報を用いる。CRF では、IOB 形式のタグがついた入力単語列を学習し、各単語のスロット種別を推定する。

3.2 応答文候補生成

応答文候補生成では、予め相槌と質問文を用意する。

まず、相槌は、ユーザの発話の後に応答するもので、表 6 に一覧を示す。相槌は表 6 全て応答文候補として生成される。

続いて、質問文について説明する。ユーザの行動をス

表 7: スロット種別に対応する質問文一覧

スロット種別	質問文
initial	今日は何をしたの?
place	店の名前はなに? どこへ行ったの? 学校のどこへ行ったの?
person	誰とそこへ行ったの?
emotion	どうだった? 料理はどんな味だった?
traffic	そこまでどうやって行ったの?
kind	そこで何を買ったの? そこで何を食べたの?
event	そこで何をしたの? そこで何をして遊んだの?
price	何円だった?
date	何時間遊んだの?

ロット種別に基づいて質問する。抽出したいスロット種別に対応する質問文の一覧を表 7 に示す。initial について、この質問文はユーザとの対話の一番初めに質問する文である。place のスロット属性値を聞き出す際、ユーザの話すエピソード種別が daily_store, non_daily_store, food の時に「店の名前はなに?」という質問をする。エピソード種別が school の時は、「学校のどこへ行ったの?」という質問をする。それ以外は、下段の質問をする。emotion のスロット属性値を聞き出す際、ユーザの話すエピソード種別の話が food の時のみ、「料理はどんな味だった?」という質問をして、それ以外は、上段の質問をする。kind のスロット属性値を聞き出す際、ユーザの話すエピソード種別が food の時のみ、「そこで何を食べたの?」という質問をして、それ以外は、上段の質問をする。event のスロット属性値を聞き出す際、ユーザの話すエピソード種別が play の時のみ、「そこで何をして遊んだの?」という質問をして、それ以外は、上段の質問をする。

表 7 から質問文の候補を生成する。スロットの属性値が空の状態のスロット種別について、その属性値を聞き出すような質問全て質問文の候補として生成する。

3.3 応答文選択

応答文選択では、生成された相槌候補と質問文候補の中から、ランダムに選択することで応答文を出力する。相槌について、生成された相槌 (表 6) の中からランダムに一つ選択する。今回は、相槌で評価に影響を与えないように、ポジティブな応答に統一した相槌を行う。質問文は、スロット属性値が空の状態のスロット種別を聞き出すような質問文の候補が応答文候補で生成されており、その中からランダムに質問文を選択する。選択された相槌と質問文を組み合わせることで、応答文を出力する。

表 8: スロットの属性値抽出の流れ 1

place	person	emotion	traffic	kind	price
-	友達	-	-	とんかつ	-

表 9: スロットの属性値抽出の流れ 2

place	person	emotion	traffic	kind	price
-	友達	-	-	とんかつ	800 円

3.4 対話の流れ

以下の対話例を用いて、ユーザ発話文からシステム発話として出力するまでの流れを説明する。各発話の番号は説明の番号に対応している。

システム 今日は何をしたの? (1)

ユーザ 友達ととんかつを食べに行ったよ (2)

システム うんうん! 何円だった? (3)

ユーザ 800 円だったよ (4)

システム ええ! 料理はどんな味だった? (5)

- (1) 最初に initial の質問として「今日は何をしたの?」という質問を生成し選択する。
- (2) ユーザの発話に対して、システムは発話理解としてエピソード種別の分類を行う。ユーザの発話はエピソード種別 food に分類される。また、ユーザの入力文中の単語からスロットの属性値を抽出する。対話例の発話では、person に「友達」、kind に「とんかつ」という属性値が抽出される。システムは、抽出されたスロット種別を表 8 のようにフレームに保存する。
- (3) 表 8 では、place, emotion, traffic, price のスロット種別がまだ埋まっていない。よって、システムはこの 4 つの属性値を聞き出す質問を応答文候補として生成する。応答文選択として、4 つの質問からランダムに 1 つのスロット種別を選択し、選択された質問文を出力とする。対話例においては、price のスロット種別がランダムに選択されたため、price のスロット属性値を聞き出す質問文である「何円ですか?」が出力された。
- (4) ユーザの応答から price の属性値「800 円」が抽出されたので、表 9 のように price の属性値を埋める。
- (5) システムは、まだ埋まっていないスロットの属性値を埋めるような質問候補を生成し、その中からランダムに選択する。対話例においては、emotion のスロット属性値を埋める質問が選択された。

3.5 対話の終了

対話の終了条件として、place, person, emotion の 3 つのスロット種別が埋まった段階で、対話を終了することとした。もし、毎回スロット種別を全て埋める対話を行うと、この対話システムを使っていくうちに、ユーザはこのエピソード種別の話をするとこの質問事項が聞かれるということを知ってしまう。質問事項を知ってしまうと、対話が事

表 10: 実験参加者から収集した日記

日付	日記	感情極性
2021/10/01	今日の清掃バイトはとても疲れた	negative
2021/10/02	今日はスーパーでシャンプーを買った	even
2021/10/03	美容院で髪を切ってもらいさっぱりした	positive

務的になり楽しくないのではないかと考えた。よって、ライフログを収集するうえで重要な 3 つのスロット種別のみ毎回質問し、それ以外は毎回質問しないように実装した。

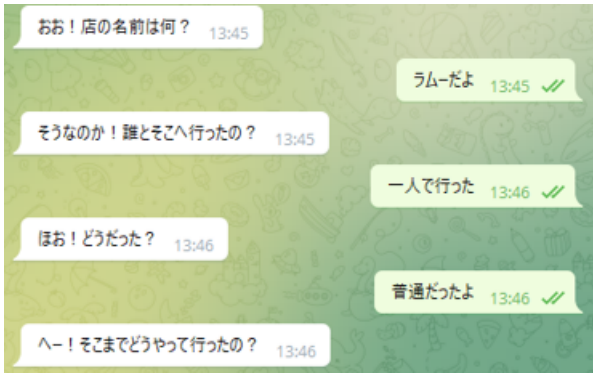
4. 評価実験

実装した対話システムにおいてユーザの対話満足度の評価を分析するために、2 つの対話システムを使用し、各システムと対話し、評価を行う。実験参加者は 4 人である。比較する 2 つの対話システムは Pronoun-DS と Estimated-DS である。具体例を図 3 に示す。図 (3a) の Pronoun-DS(以降、P-DS と記す) では、スロット種別が抽出されていて、スロット種別の place が埋まっていたとしても「そこへ」や「そこまで」のように指示代名詞を使って対話する。この方法での対話は、スロット種別に対応する属性値の抽出が誤っていても気づかれにくく、対話に一貫性があると感じられる一方、システムに、ユーザの話を聞いてもらっていると感じにくく、対話が楽しいと感じられないというデメリットがあると考えられる。図 (3b) の Estimated-DS(以降、E-DS と記す) では、スロット種別に対応する属性値が抽出されている場合、その属性値を使った質問を行う対話システムである。この方法での対話は、システムに話を聞いてもらっていると感じやすい一方、スロットの属性値抽出を誤ると評価に大きく影響するのではないかと考えられる。

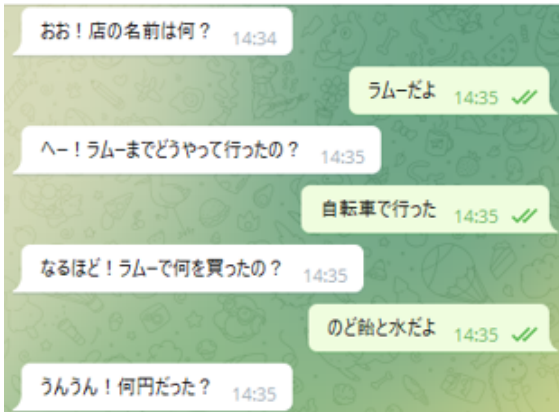
4.1 実験準備

学習データにユーザ毎の特徴を反映させるため、実験参加者 4 人 (A~D) に毎日一文ずつ日記を書いてもらった。また、日記の行動に対する感情がポジティブなのか、ネガティブなのかどちらでもない (even) なのかという感情極性も記録してもらった。実験参加者 B の日記内容の一部を表 10 に示す。収集した文の数は、実験参加者 A が 114 文、実験参加者 B が 78 文、実験参加者 C が 23 文、実験参加者 D が 24 文である。この日記から単語を手作業で抽出しスロット種別の単語ファイルに追加し、学習データを作成する。

実験に用いたエピソードは、各エピソード種別に対して 1 つずつであり、これらは各実験参加者に日記の中から選択してもらった。エピソードの具体例を表 11 に示す。実験参加者は準備したエピソード (表 11) を基に、全てのエピソード種別について一つずつシステムとやりとりする。



(a) Pronoun-DS



(b) Estimated-DS

図 3: 対話システムの対話例

表 11: エピソード例

エピソード種別名	エピソード (スロット種別ごとに収集)				
	place	person	emotion	traffic	event
daily_store	イオン	1人	普通 (even)	歩き	-
non_daily_store	美容院	友達	楽しい (positive)	車	散髪
food	焼肉屋	妹	美味しい (positive)	バス	-
play	兵庫県	友達	面白い (positive)	電車	スキー
school	図書館	1人	疲れた (negative)	自転車	勉強
work	バイト	後輩	しんどい (negative)	自転車	掃除
home	-	1人	退屈 (negative)	-	ゲーム

4.2 実験手順

実験参加者は、まず、比較する対話システムとは異なる相槌のみを返す対話システムと対話し、telegram の操作の練習を行う。続いて、一方のシステムと1つのエピソードについて対話を行う。1つのエピソードについて話し終わった後、1エピソードの対話ごとに評価を行う。評価項目については4.3項で述べる。7つのエピソードの対話と評価を繰り返す。一方のシステムの7エピソード全ての対話と評価を終えると、もう一方の対話システムについて7エピソードの対話と評価をする。また、エピソードの順番はランダムとする。

4.3 評価項目

主観評価の評価項目として、1ターン毎の評価と、1エピソード毎の評価を行った。



図 4: 1 ターン・1 エピソード毎の評価

表 12: 1 ターン評価項目

番号	項目
q1	システムの質問は現在の話と矛盾していないか
q2	システムに話を聞いてもらったと感じたか

表 13: 1 エピソード評価項目

番号	項目
q3	会話は楽しかったか
q4	もう一度話したいか
q5	会話に一貫性はあったか

1 ターン評価は、ユーザの発話と次のシステムの発話を1ターンとして、1ターン毎に評価を行ってもらった。具体例を図4に示す。図4の右側がユーザの発話、左側が対話システムの発話である。図4のオレンジの枠で囲まれた部分を1ターンとする。評価項目は、表12である。

1 エピソード評価は、1つのエピソードの話が終わるまでを1エピソードとし、1エピソードごとに評価を行ってもらった。図4の青色の枠で囲まれた部分を1エピソードとする。評価項目は、表13である。

q1~q5の評価項目について、「全くそうは思わない」、「そうは思わない」、「ややそうは思わない」、「どちらでもない」、「ややそう思う」、「そう思う」、「とてもそう思う」という7段階のリッカート尺度で評価する。結果の分析の際には、「全くそうは思はない」から「とてもそう思う」まで、それぞれ1から7の数字を割り当て、間隔尺度として分析する。

評価の考察の際には、仮説1「エピソード種別やスロット種別の推定がともに正解した際は、ユーザにとって、E-DSの方が話を聞いていると感じられ、楽しい・また使いたいと感じる」と、仮説2「エピソード種別やスロット種別の推定が誤った場合、P-DSの方が推定誤りの影響は少ない」という2つの仮説をもとに考察する。ここで、1エピソード

表 14: 各ユーザのターン数

システム	A	B	C	D	平均ターン数
P-DS	25	31	28	31	28.75
E-DS	31	37	29	31	32

表 15: エピソード種別とスロット種別ともに正解したエピソード数

システム	A	B	C	D	平均エピソード数
P-DS	3/7	5/7	5/7	3/7	4 エピソード
E-DS	3/7	5/7	4/7	3/7	3~4 エピソード

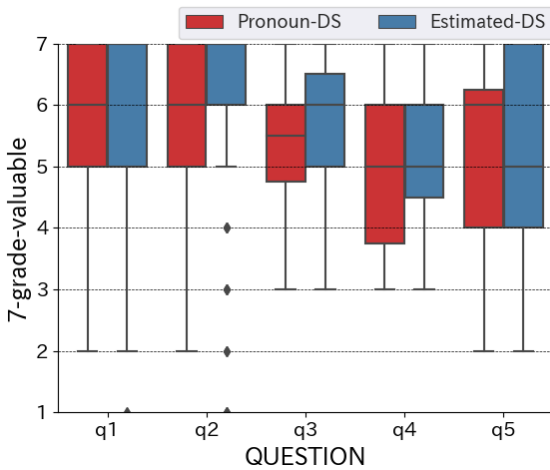


図 5: 2つのシステムの評価 (エピソード種別とスロット種別ともに正解の場合)

ドの中の全てのターンのエピソード種別とスロット種別の推定結果が全て正解であれば、『エピソード種別とスロット種別がともに正解』と定義する。

4.4 評価結果

表 14 にユーザのターン数を示す。横軸の A~D は各実験参加者、縦軸がそれぞれのシステムで、各実験参加者がそれぞれのシステムについて合計何ターンしたかを示している。スロット種別とエピソード種別の推定結果が、実際に日記に書いてもらった単語 (表 11) と一致していたら正解とする。

エピソード種別とスロット種別がともに正解したエピソード数を表 15 に示す。

4.5 仮説 1 の結果

仮説 1 を検証するために、2つの比較を行った。

エピソード種別とスロット種別共に正解の場合について、P-DS と E-DS を比較した (図 5)。q1, q2 は 1 ターン評価で q3~q5 は 1 エピソード評価である。図 5 の q2 に関して、中央値は P-DS, E-DS ともに 6 だが、ばらつきは E-DS の方が小さいことが分かる。また、最小値が E-DS の方が高い。よって、エピソード種別とスロット種別がともに正解

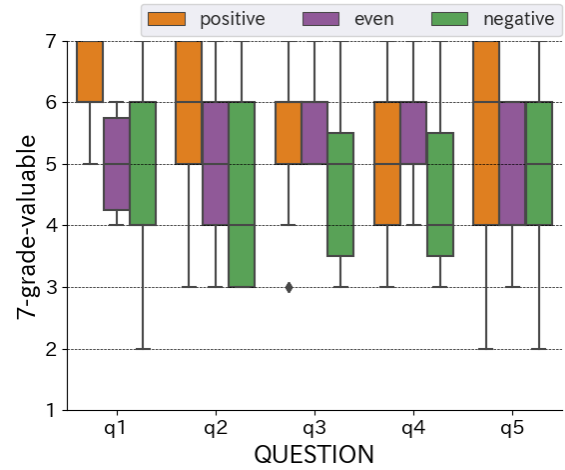


図 6: 感情極性ごとの評価 (エピソード種別とスロット種別ともに正解の場合)

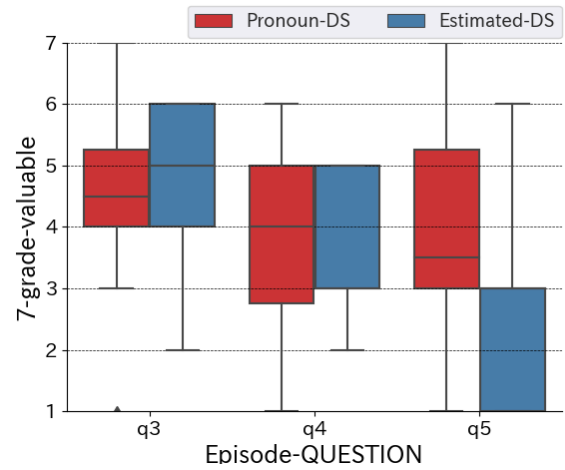


図 7: 2つのシステムの 1 エピソード評価 (エピソード種別とスロット種別不正解の場合)

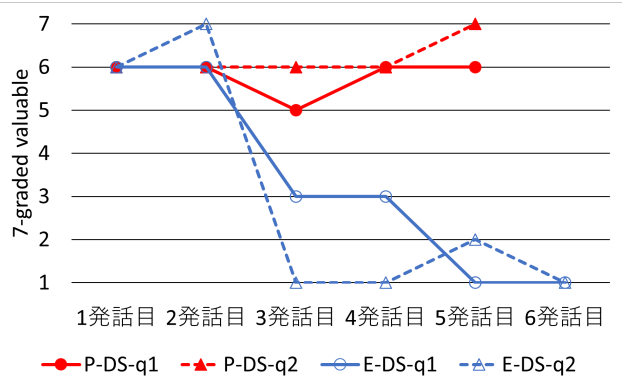


図 8: 1 ターン評価の推移。P-DS, E-DS ともに 3 ターン目でスロットの属性値抽出誤り

の場合、聞いてもらっていると感じられるのは E-DS の方であるといえるが、大きな違いではないことが分かる。図 5 の q3, q4 に関して、最大値や最小値、中央値が P-DS と E-DS ともにほぼ同じ値を示している。よって、楽しい・

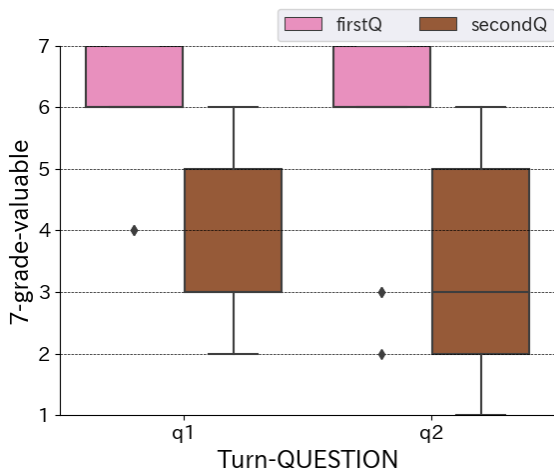


図 9: 同じ質問があるかどうかの 1 ターン評価

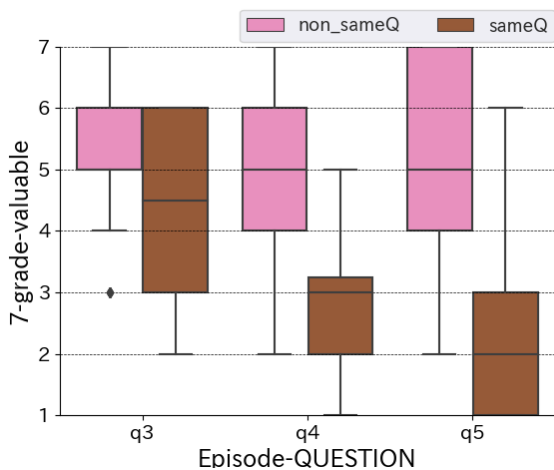


図 10: 同じ質問があるかどうかの 1 エピソード評価

また使いたいという評価において、2つのシステムに大きな差はないことが分かる。

次に、エピソード種別とスロット種別がともに正解の場合、どのような条件で評価に違いが出るのかを検証するために、ユーザの感情極性ごとの評価について比較した。

エピソード種別とスロット種別共に正解の場合について、ユーザの話すエピソードをユーザの感情極性ごとに比較した(図6)。図6について1ターン評価(q1, q2)では、ユーザの感情発話とシステムの応答の組に着目し、この1ターンを感情極性ごとに評価したものである。感情発話とは、システムがemotionスロット種別を埋めるような質問をした際に、ユーザが応答する「楽しい」や「疲れた」などの感情を表す発話のことである。1エピソード評価(q3~q5)では、1エピソードについて感情極性ごとに評価したものである。図6のq1, q2に関して、どちらも中央値がpositive, even, negativeの順に小さくなっていることが分かる。また、negativeのばらつきが大きいことが分かる。図6のq4, q5に関して、ネガティブの四分位範囲が他二つに比べて低いことが分かる。よって、1ターン評価、1エ

ピソード評価ともに、感情極性がポジティブなほど評価が高く、ネガティブなほど評価が低いことが分かった。

4.6 仮説2の結果

仮説2を検証するために、3つの比較を行った。

エピソード種別とスロット種別いずれかに誤りがある場合について、P-DSとE-DSを比較した。この比較についての1エピソード評価を図7に示す。図7のq5に関して、P-DSに比べて、E-DSは最大値や四分位範囲、中央値が大きく低くなっていることが分かる。また、q3の中央値や四分位範囲がE-DSの方が高くなっているが、q4では中央値が低くなっていることを考察すると、一度の使用であれば楽しいと感じるが何度も使い続けたくないという結果であると考えられる。このように、エピソード種別とスロット種別いずれかに誤りがある場合、E-DSの方が評価が下がった理由について、さらに2つの比較を行った。

図8は、ある実験参加者の1ターン評価の推移をP-DS, E-DSの1ターン評価ごとに折れ線グラフで示したものである。どちらもfoodエピソード種別について話している。P-DSは、3ターン目で、本来の飲食店名「キャナリーロウ」ではなく、「キャナリーロウっていうん」のようにplaceスロットの属性値抽出を誤り、4ターン目で「そこまでどうやって行ったの?」、5ターン目で「誰とそこへ行ったの?」のような発話を行った。E-DSは、3ターン目で、「洋風な味」というemotionの属性値をplaceの属性値として抽出し、4ターン目で「洋風な味までどうやって行ったの?」、5ターン目で「誰と洋風な味へ行ったの?」のような発話を行った。図8のP-DSの3ターン目ではq1の評価が少し下がったがそのあと評価は上がっている。一方で、E-DSでは4ターン目からq1, q2とも評価が大きく下がっている。

また、同じ質問があるかどうかの評価を図9と図10に示す。システムがスロットの属性値を抽出できなかった際、1エピソード中に同じ質問を複数回繰り返すことがあった。例えば、emotionの属性値を抽出できなかった際、何度も「どうだった?」と質問することがあった。図9の1ターン評価では、複数回同じ質問が繰り返されたエピソード中で、1回目の質問(firstQ)と2回目の質問(secondQ)の二つで比較した。図10の1エピソード評価では、エピソード中に同じ質問が複数回でてきたエピソード(sameQ)と、同じ質問は出てこなかったエピソード(non-sameQ)の2つで比較した。

4.7 考察

4.7.1 仮説1の考察

仮説1に関して考察する。4.5項から、エピソード種別とスロット種別の推定結果がともに正解の時、ユーザの感情極性によって評価が変わることが分かった。この原因として、現状の対話システムはユーザのどのような応答にも

ランダムで相槌を返すようになっている。この相槌はポジティブな応答が多い。よって、ネガティブな話をしているときにポジティブな応答で返すため、ユーザの評価が下がったのではないかと考える。実験参加者の記述式のアンケートでも、「ネガティブな話をしているときにシステムの応答のテンションが高いことが気になった」という声があった。

これらの結果から、今後の対話戦略として、感情極性ごとに応答を変える必要があると考えられる。具体的には、収集されたライフログとユーザが話しているエピソードから、ユーザが話しているエピソードは「楽しい」のか「悲しい」のか等の感情を推定する。さらに、その感情から感情極性を推定し、感情極性がポジティブの場合は積極的に質問し、感情極性がネガティブの場合はユーザを労わる応答をする等、感情極性ごとに相槌や応答を変えることが必要であると考えられる。

4.7.2 仮説2の考察

仮説2に関して考察する。4.6項から、仮説通りエピソード種別とスロット種別いずれかに誤りがあった場合、P-DSの方が推定誤りの影響は少ないことが分かる。ここで、エピソード種別とスロット種別いずれかに誤りがあった場合でE-DSの評価が下がった原因を2つ考察する。

1つ目の原因として、P-DSはスロットの属性値抽出誤りがあっても抽出された単語を使わない為、気づきにくい一方、E-DS抽出された単語をそのまま使うため、明らかに間違いだと気づきやすいことがあげられる。4.6項の図8からもわかる通り、抽出された単語を使うかどうかで評価に影響が出ることが考察できる。

2つ目の原因として、エピソード中にシステムが同じ質問を複数回繰り返すことがあげられる。4.6項の図9と図10から2回以上同じ質問があったエピソードでは評価が下がっていることが分かる。ここで、P-DSでは5ターン、E-DSでは9ターン同じ質問を行っていた。よって、E-DSの方が同じ質問をした回数が多いため評価が下がったとも考えられる。

これらの結果から、今後の対話戦略として、スロットの属性値の推定確率を使用した方法を検討する。スロットの属性値の推定確率が低い場合は、指示代名詞を用いた質問にすることで、スロットの属性値抽出が誤っていても気づかれにくくなるを考える。また、同じ質問をする際は、別の表現で質問するなどの戦略が必要だと考えた。

5. まとめ

本報告では、発話を変えることでユーザに親密さを感じさせる対話システムを目指し、その為の前段階として、ユーザのライフログを抽出する対話システムを作成した。この対話システムを用いてユーザの評価を行った。

実験結果から、推定結果が正解の際には、ユーザの感情

極性ごとに発話内容を変える必要があることが分かった。また、推定結果が不正解の際には、誤ったスロット種別を用いた応答をすると一貫がないように感じられることが分かった。

今後の対話戦略として、収集されたライフログから、ユーザが話そうとしているエピソードに対する感情を早期に推定し、推定された感情極性を活かした応答をすることで、ユーザに親密さを感じさせる対話システムが実現できると考えられる。また、スロット種別の推定確率が低いときは指示代名詞を用いた発話を行い、推定確率が高いときは抽出されたスロットの属性値を用いた発話を行うなど、推定確率によって発話を変えることで、前者の場合は、推定結果が不正解の場合のリスクを軽減でき、後者の場合は、ユーザがシステムに話を聞いてもらっていると感じる対話システムになると考えられる。

参考文献

- [1] Xianchao Wu, Kazushige Ito, Katsuya Iida, Kazuna Tsuboi, and Momo Klyen. りんな: 女子高生人工知能. 言語処理学会第22回年次大会発表論文集, pp. 306–309, 2016.
- [2] Amit Kumar Pandey and Rodolphe Gelin. A mass-produced sociable humanoid robot: Pepper: The first machine of its kind. *IEEE Robotics & Automation Magazine*, Vol. 25, No. 3, pp. 40–48, 2018.
- [3] Dennis Perzanowski, Alan C Schultz, William Adams, Elaine Marsh, and Magda Bugajska. Building a multimodal human-robot interface. *IEEE intelligent systems*, Vol. 16, No. 1, pp. 16–21, 2001.
- [4] 前蘭そよぎ, 原直, 阿部匡伸. 人対人の会話で自然な話題展開を支援するための対話戦略の検討. 情報処理学会研究報告, Vol. 2021-SLP-137, No. 16, pp. 1–6, 2021.
- [5] Shunsuke Tada, Yuya Chiba, Takashi Nose, and Akinori Ito. Effect of mutual self-disclosure in spoken dialog system on user impression. In *2018 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, pp. 806–810. IEEE, 2018.
- [6] 三上聡美, 山口裕幸. 親密度の異なる友人に対する自己開示抵抗感に関する検討. 九州大学心理学研究, Vol. 9, pp. 75–81, 2008.
- [7] 川島嵩弘, 西村隆志, 安江駿亮, 六沼元貴, 和田史織, 杉本徹. ライフログ雑談対話システムに関する研究. 人工知能学会研究会 言語・音声理解と対話処理研究会資料, Vol. 78, pp. 120–121, 2016.
- [8] 角森唯子, 東中竜一郎, 吉村健, 磯田佳徳. ユーザ情報を記憶する雑談対話システムの構築とその複数日にまたがる評価. 人工知能学会論文誌, Vol. 35, No. 1, pp. DSI-B-1–10, 2020.
- [9] telegram. <https://telegram.org/>.
- [10] 東中竜一郎, 稲葉通将, 水上雅博. Python でつくる対話システム. 株式会社 オーム社, 2020.