

自らの体験に基づき雑談する対話システム

Chat-Oriented Dialogue System Based on Its Own Experience

成松宏美* 杉山弘晃 水上雅博 有本庸浩
Hiromi Narimatsu Hiroaki Sugiyama Masahiro Mizukami Tsunehiro Arimoto

NTT コミュニケーション科学基礎研究所
NTT Communication Science Laboratories

Abstract: Recent research on chat conversation has led to the development of dialogue systems that can respond to users over a wide range of topics, which was the first major challenge of chat conversation. However, it is still difficult to construct dialogue systems that can properly respond to user utterances according to the dialogue context, and this has often made users feel that the system did not understand what they said. In this paper, we propose a chat dialogue system that converses based on its own experience, in order to make users feel the ability to understand the dialogue context. This system is submitted to the Dialogue System Live Competition.

1 はじめに

スマートフォン向けの対話エージェントやAIスピーカなどの普及に伴い、対話システムが一般家庭で使われるようになってきた [7]. これらの対話システムは、“明日の天気を教えて”や“今日のニュースを教えて”などのユーザの質問に回答することを目的としたタスク指向型対話と呼ばれるが、明確なタスクが決められていない雑談対話システムにも注目が集まっている [5]. 従来より、観光案内などのタスク対話システムに対して、雑談のような声かけをすることが知られており [9], 一般家庭で使われているAIスピーカにおいては、一般家庭で使われるAIスピーカに対しても、同様の声かけが見受けられることが想像できる.

こうした需要の増加に伴い、会話を楽しむことを目的とした雑談対話システムの研究も増えてきている [10]. 雑談対話では、ユーザが対話システムに対してどんな声かけをするかの予測ができないことから、ユーザの幅広い話題に関する発話や質問に対して応答できるよう、多量の発話ペアをニューラルネットワークで学習し、システムの発話を生成または選択する一問一答ベースの手法が成功を収めている [10, 2].

一方で、一問一答ベースの対話システムでは、対話履歴を見ていないため、対話の文脈に整合しない発話をする事が知られている [12]. 文脈に矛盾しない発話をするために、対話履歴を入力として応答を生成する手法もあるが [11], 現状では、雑談のような対話データを多量に収集することが難しいことから、現状では話

システム:	夏休みはどうだった？
ユーザ:	広島に行ってきたよ！
システム:	いいね！厳島神社は行った？
ユーザ:	うん！行ったよ！
システム:	お！行ったんだね！景色綺麗だよな？
ユーザ:	うん！すごく綺麗な景色だった！
システム:	だよな！私も行ったんだけど、海の中の鳥居がすごく綺麗でよかった！
ユーザ:	うんうん！ (このシステムわかってくれてる！)

図 1: 提案する対話システムとユーザの対話例

題の制約として文脈を利用することにとどまっている.

そのため、依然として文脈に整合しない発話をする事は多々あり、こうした発話が複数回起きると、ユーザは「このシステムはちゃんと話を聞いてくれてるのかな？」とシステムの理解能力に疑問を感じてしまい、嬉しいことや悲しいことを共有して共感してもらいたいときには、対話を通して満足感を得ることが難しい.

我々は、相手が人であるときと同じように、文脈を理解し、対話を通してユーザの気持ちを理解できる相手であるとユーザに感じてもらえるような対話システムの実現を目指し、研究を行っている [13]. 特に、気持ちを理解してくれることは、対話の満足度において重要とされている. そこで、本稿では、ユーザとシステムの相互の体験に特化して、ユーザの話した体験などを文脈情報として構造化して理解し、理解したことをシステム自身の体験に紐付けて質問や共感を行う手法

*連絡先: NTT コミュニケーション科学基礎研究所
619-0237 京都府相楽郡精華町光台 2-4
E-mail: hiromi.narimatsu.eg@hco.ntt.co.jp

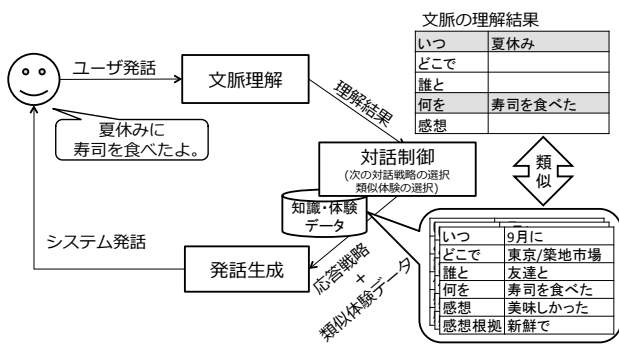


図 2: 提案するシステムの概要。

を提案し、「理解している」という態度をユーザに示す効果を狙う。具体的には、図 1 に示すように、5W1H+感想のフレームをユーザの体験として引き出す対話戦略を用いて、ユーザの発話に含まれる 5W1H+感想に相当するフレーズを正しく抽出し、同様の形式で擬似的に作成したシステムの体験データと照合し、ユーザの体験に類似したシステムの体験を用いて質問や共感を行う。対話システムがユーザの話を理解しているという態度を示す方法としては、ユーザ発話の一部を繰り返して、質問や共感を行う手法が有効であることが従来より知られている [6]。これに対して、提案手法では、ユーザから引き出した体験をシステムの体験に紐付けて質問や共感を行うことができるため、システムがどのように捉えたのかの理解を、ユーザに暗に伝える効果が生まれ、「理解している」という態度が感じられる対話になると考えた。

実際に提案手法と従来手法の対話を実装し、人とのテキストチャットにより評価を行った結果、提案手法の方がシステムの印象（ここでは「理解している」かどうか）が向上することがわかった。また、本システムを、対話システムライブコンペティション 2 [14] のシチュエーショントラックに提出し、予選通過となった。シチュエーショントラックは、友人と過去に行った旅行について対話をするという設定で、クラウドワーカーが、本システムとテキストチャットをして、「それくらい（そのシチュエーションに適した）人らしいか」を評価する。本稿では、予選で得られた評価結果およびアンケートの内容について紹介する。

2 提案システム

ユーザがシステムに対して「理解してくれている」と感じてもらえる対話システムの実現を目的として、対話を通してユーザからシステムに伝えられるユーザの体験および気持ちを文脈として正しく理解し、理解した結果をシステムの体験に紐付けて質問や共感を行う

手法を提案する。具体的には、対話を通してユーザから体験を引き出す対話に着目し、図 2 に示すように、ユーザの体験を表す 5W1H+感想を文脈として理解し、同様の構造でシステムの体験を予め記述しておくことで、ユーザの体験に類似したシステムの体験に基づく関連質問および共感を行う。体験を引き出す際の対話の流れについては、自然な対話の流れに沿ったアジェンダベースのシステム [12] をベースとして設計する。以下に、対話戦略、文脈としてユーザの体験を理解する方法、およびシステムの体験に基づく発話生成手法について述べる。

2.1 対話戦略

対話を通してユーザの体験を引き出すために、5W1H の質問をしていく戦略を採用する。5W1H のフレームは非常にシンプルであるが、体験を構造化して表現する際の重要な項目であることに加えて、いろいろなシチュエーションで共通して使えるフレームである。たとえば、人とのコミュニケーションやカウンセリングを目的とした対話システムにおいても 5W1H の質問をしていく戦略がとられており [4]、対話システムにも採用されているだけでなく、人同士が旅行の話をしたり、美味しいものを食べた話をしたときにも、「どこに行ったの?」「いつ行ったの?」「どうだった?」などは自然な流れで出てくる質問であると言える。そこで、本フレームを参考に、「いつ」「どこで」「誰と」「何を」「感想」をユーザの体験として理解する項目として設定し、これらの項目を引き出す質問をしていくような対話戦略でシステムを設計する。

しかし、この戦略だけでは、すでにユーザが話したことを質問してしまう可能性や、なぜこのシステムがそれを質問したかがわからないまま、あれこれ質問される対話になる可能性がある。これに対して、質問したこと以外も含め、ユーザの発話から正しく 5W1H+感想の項目を理解する方法と小さな対話の流れにも評価やなぜ質問したのかがわかるような自己開示を含める対話戦略 [12] を用いて、自然につながる対話を実現する。

2.2 文脈の構造化理解

システムの体験との照らし合わせが容易な、ユーザの体験を対話履歴から文脈として理解するために、体験を構造化して理解する手法を提案する。たとえば、対話中にユーザが「夏休みに寿司を食べたよ。」「築地に行ってきたよ。」という発話をした場合に、2 つ発話を関連付けて、{ いつ:夏休み, どこで:築地, 何をした:寿司

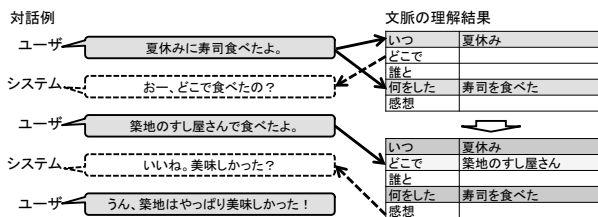


図 3: 文脈理解の様子.

を食べた}と、理解する。こうすることにより、5W1Hの情報を質問していく対話戦略において、すでにユーザが話したことを質問する可能性を抑制することができる。

しかしながら、5W1Hに該当するフレーズを発話から抽出することは、これまで明確な需要がなく、あまり取り組まれていなかった。そのため、たとえば「家の近くのスパで岩盤浴をしたよ。」という発話が入力された場合には、{どこで:家の近くのスパ, 何をした:岩盤浴をした}の2つの項目を抽出することが正しいが、固有表現ではない一般名詞の組み合わせであるフレーズ「家の近くのスパ」を「どこで」に該当するフレーズとして抽出することは難しい [8]。

そこで、固有表現抽出と同様に系列ラベリングの技術を用いて、該当するフレーズにラベル付けしたデータを新たに学習する。こうしたフレーズを抽出するための学習データはないため、人手で作成する必要がある。そこで、人手での学習データ作成の手間を考慮し、システムの擬似体験として記述した5W1H+感想のデータを用いて、ルールかつ自動的に、多様な組み合わせの発話例および抽出対象フレーズへのラベル付けを行い、フレーズ抽出器を学習する。

このように構築したフレーズ抽出器を用いることで、雑談に含まれるような口語表現であっても、5W1H+感想のフレーズを検出できるようになり、ユーザの体験を5W1H+感想として理解することが可能となる。結果として、図3に示すような、文脈に沿った質問生成を実現できる。

2.3 システムの体験に基づく発話生成

理解したシステムの体験と、システムに擬似的に設定した体験データを用いて、関連質問および共感発話を生成する。前述の構造化理解によって、「夏休みに旅行に行った」と「大阪観光した」という発話から、{いつ:夏休み, どこで:大阪}を一つの体験に紐付けて理解できる。そのため、2つの情報を関連付けて、システムの体験から類似の体験を探ことができ、{いつ:夏休み, どこで:大阪}が合致するシステムの体験に{どこで:大阪/海遊館, 何をした:魚を見た}があれば、「海

遊館とか行った?」という関連する質問を生成したり、{どこで:大阪/道頓堀周辺, 何をした:散歩した, 感想:暑かった}「夏の大阪は暑いよね」という発話を生成する。これにより、文脈に整合した発話の生成が期待できる。

さらに、気持ちを理解していることを示すために、図4に示すような発話生成を行う。たとえば、「私も[どこで][何をした]よ。」や「[感想の理由][感想]よ。」といったテンプレートを作成し、システム自身も同じような体験をしたことがあることを提示した上で、ユーザの気持ちに共感したり、そのような気持ちになった理由を説明する。これにより、ユーザが、あたかもシステムが自ら体験して共感してくれているように感じることを狙う。システム自らが体験してユーザと似たような感想を抱いたという印象を、ユーザに対して与えることができれば、暗にユーザの気持ちを理解できる能力があることを示せることが期待できる。

3 予備実験

3.1 準備

提案システムを実装する上で必要となる、データやフレーズ抽出器の作成方法について以下に説明する。今回は、人同士の雑談において体験や感想の共有が発生しやすい、旅先での体験に特化して、対話システムの設計およびシステムの擬似体験データを作成した。体験データは、「いつ」「どこで」「誰と」「何をした」「感想」に加えて「感想の理由」を一つの構造化されたまとまりとして、日本全国の各都道府県の有名な観光地に行った場合を想定して、人手で作成した。全部で2653個の体験データを作成した。

次に、フレーズ抽出器の構築方法について述べる。フレーズ抽出器の学習データ作成にかかる手間を低減するために、上記で作成した体験データから自動的に学習用データを作成する。たとえば、「[どこで][何をした]。[感想]。」や「[いつ][どこで][何をした]。[感想理由][感想]。」などのいくつかのテンプレートを用意して、各項目にデータを代入することで、学習に用いる発話文を生成する。テンプレート代入と同時に、検出対象となる区間に、BIOタグを付与する。抽出器には、BERT [3]の系列ラベリング手法を用い、作成した学習用データを用いて、系列ラベリング手法を転移学習する。

3.2 体験に基づく発話の効果

提案システムを実装し、システムの体験に基づいた発話の有効性を評価する。気持ちをわかってもらえたと思うことは重要であることから、共感発話の部分に

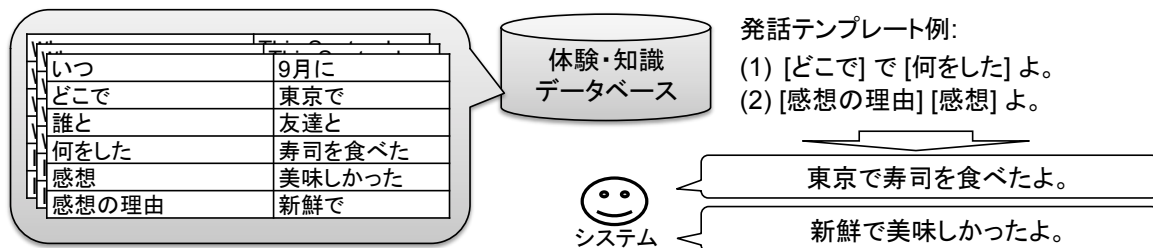


図 4: システムの体験データを用いた発話生成の例。

ついて、システムの体験に基づいた発話生成がどれくらい「理解してくれている」と感じる効果を向上できるかを評価する。具体的には、提案手法と同様の対話戦略をベースとして構築する以下3つのシステムを実装し、比較する。

- **復唱:** ユーザ発話の一部を繰り返す、シンプルな共感発話「いいね。」や、「有名だよね。」など、システムの体験と紐づかない発話をする。
- **自体験:** 提案手法のテンプレートを用いて、自らの体験として「私も[何をした]よ。[感想]よね。」や、「私も[何をした]よ。[感想の理由][感想]よ。」などを発話する。
- **伝聞体験:** 提案手法のテンプレートを用いて、伝聞体験「ネットで[何をした][感想]という情報を見て行ってみたいと思った。」や、「友達が[どこで]行って。[感想の理由][感想]っておすすめしてた。」などを発話する。

体験を話す場合には、自分の体験として話す場合と伝聞体験として話す方法が考えられる。いずれにおいてもシステムの体験（自らが体験したこと、自らが見たこと、および聞いたこと）であると言えるが、印象に違いがあるかを確認するため、自体験と伝聞体験の2パターン作成し、比較する。

前述の3つのシステムを用いて、対話システムがどれくらい「理解している」ように感じたかの主観評価を比較する。20代から50代のクラウドワーカー男女合計36名に、各システムと対話してもらい、「理解している」ように感じたかの評価を行う。評価には、7段階のリッカート尺度 [1] を用いる。

結果を、図5に示す。自体験および伝聞体験は、ユーザ発話の一部を復唱する戦略と比較して、有意に「理解していた」という印象が向上した(有意差は、どちらも $p=0.01$)。結果より、自体験および伝聞体験を用いてシステムの発話を生成した場合には、ユーザ発話の一部を復唱する戦略よりも、「システムは理解している」という印象を向上させる効果があることがわかった。

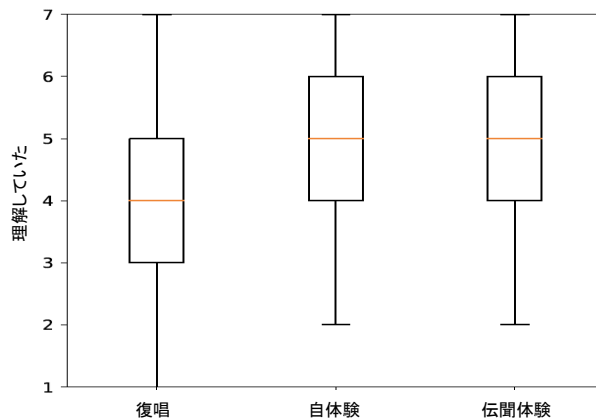


図 5: システムの体験に基づく発話が、ユーザのシステムに対する印象に与える効果。

4 ライブコンペティション2の評価

提案した対話システムを、対話システムライブコンペティション2のシチュエーショントラックに提出し、予選通過となった。シチュエーショントラックは、過去に行った旅行について友人と対話をするという設定で、クラウドワーカー50名が、本システムとテキストチャットをして、「どれくらい(そのシチュエーションに適した)人らしいか」を評価する。発話の開始が「ところで、これまで行ったところで一番印象に残った場所ってどこ?」と定められていることから、旅先での体験について対話を行う我々の提案システムは、本シチュエーションと親和性が高いと考え、シチュエーショントラックへ提出した。

以降に、予選で得られた評価結果および任意アンケートで得られたコメントの内容について紹介する。「どれくらい(シチュエーションに適した)人らしい会話か」の評価値は、3.96 (1が最高, 5が最低)で、全7システム中3位の成績であった。予選で行われた実際の対話例(図6)をみると、コメントの通り比較的自然的な対話ができていることがわかる。他のシステムとの比較については、対話システムライブコンペティション2のまとめ [14] を参照されたい。

50人中31人のワーカーからコメントを受領した。よかったコメントでは、「本当に会話している気分になった。」「人と話しているようだった。」「本当に友達と話しているようだった。」など、人との会話に近い発話がある程度できていることが伺えた。一方で、評価値が低かった対話では、「会話が噛み合っていなかった」というコメントもあった。実際にログをみると、ユーザの体験と類似したシステムの体験を選択する際に、ユーザが言及した県とは別の県の体験を話していることがあった。これはシステムの設計の以下2点に起因する。**対話履歴に基づく話題の限定**

システムは、対話の最初の方でユーザが「京都がよかったよ」と発話をした場合に、「京都」の話題に限定して次の対話を進めるようにしている。一方で、ユーザが途中で別の県の話（たとえば「清水寺もよかった、あとそういえば、兼六園もよかったな。」など）をすることも想定して、話題となっている地域の限定条件を少し緩めた設計となっている。そのため、同じ名称を持つ場所や同じ名称の名物を持つ地域が複数存在する場合に、別の地域にマッチしてしまうことが起こりうる。**名称表現の多様性**

人の表現は多様であり、略称を使うことも多く、同一のものを表す名称であっても同一であると解釈することが難しいことがある。たとえば、「東京ディズニーランド」について言及する場合、「ディズニーランドに行ったよ。」と言うこともあれば、「ディズニー行ってきた。」や「TDL行った。」や「ランド行った。」など、別名や略称を使うことも多い。

本システムでは、多様な言い回しを考慮して、同一のものに対して複数の表現を許容して照合できる設計にしている。しかしながら、「ランド行った。」や「温泉行った。」などでマッチできるように、「ランド」や「温泉」を照合の対象とすることはリスクがある。たとえば、「ランド」がつく場所には、「ディズニーランド」以外にも「サンリオピューロランド」が存在し、どちらが正しいかを容易に判断することは難しい。

上記の課題を解決する方法としては、ユーザの発話から体験を理解する際に、どれだけ正しく理解できたかの信頼度を付与し、信頼度が低い場合には確認を行う、システムの体験と照合する際に、複数の体験に合致した場合には、システムが発話生成に用いる体験を絞りこむための、質問をユーザに投げかける、などが考えられる。

5 おわりに

本研究では、ユーザとシステムが互いに体験したことを共有するようなシーンを想定し、システムがユーザの体験を対話から構造化して理解し、システム自ら

の体験に照らし合わせて質問や共感発話を生成する手法を提案した。システム自身が体験したかのような自己開示や、知識が関連付けられた質問ができるため、結果としてユーザのシステムに対する印象（「理解してくれる」）を向上することができた。本システムを対話システムライブコンペティション2のシチュエーショントラックに提出し、「どれくらい（そのシチュエーションに適した）人らしいか」の主観スコアで、5段階中3.96であった。コメントからも、本システムがある程度、人と同じように対話できていることが示された。一方で、ユーザの体験に類似したシステムの体験を話す場合には、ユーザの体験に対して適切なシステムの体験に基づいて話すことが求められる。「どこで」や「何をした」の取り違えの解消もしくは、取り違えに気づきりカバリをする対話制御などに取り組んでいく。

また、今回、システムに擬似的に設定した体験データは、人手により構築したものである。こうした体験データを、Webや人との対話を通して自動的に作成する手法にも取り組む。

参考文献

- [1] I Elaine Allen and Christopher A Seaman. Likert scales and data analyses. *Quality progress*, Vol. 40, No. 7, pp. 64–65, 2007.
- [2] Ashutosh Baheti, Alan Ritter, Jiwei Li, and Bill Dolan. Generating more interesting responses in neural conversation models with distributional constraints. *arXiv preprint arXiv:1809.01215*, 2018.
- [3] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.
- [4] Sangdo Han, Kyusong Lee, Donghyeon Lee, and Gary Geunbae Lee. Counseling dialog system with 5w1h extraction. In *Proc. of the SIGDIAL 2013 Conference*, pp. 349–353, 2013.
- [5] Ryuichiro Higashinaka, Kenji Imamura, Toyomi Meguro, Chiaki Miyazaki, Nozomi Kobayashi, Hiroaki Sugiyama, Toru Hirano, Toshiro Makino, and Yoshihiro Matsuo. Towards an open-domain conversational system fully based on natural language processing. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pp. 928–939, 2014.

システム: ところで、これまで行ったところで一番印象に残った場所ってどこ？

ユーザ: 沖縄かな～。

システム: おー、いいねえ。どこ観光したのー？ちょっと、どこか旅行行こうかと思っててき。

ユーザ: 美ら海水族館良かったよ！

システム: 美ら海水族館！どんな感じ？

システム: ジンベイザメ

システム: おお～！職場の人が行って、思っていたより大きかったみたいで。ちょっと気になってたの！

ところで、沖縄は、いつ行ったのー？

ユーザ: 5年くらい前かな！

システム: そうかあ。実は、夏が候補なんだけど、夏だと暑いかなあ？

ユーザ: 暑いと思うよ。10月頃だとちょうどいいかも！

…(中略)

システム: そっかあ。沖縄でいうと、名物八重山そばとか有名らしい？食べた？

ユーザ: 食べてないな。初めて聞いた！

システム: 職場の後輩がね、コクがあるスープが細麺に合っていたので美味しかったですよ！

ってめっちゃオススメしてたから聞いてみた笑。

…(中略)

システム: あと、沖縄だとひめゆり平和祈念資料館とか聞いたことある！行った？

ユーザ: 美ら海しか行ってないなあ～。

システム: そうなのね～ネット見て「沖縄戦線の歴史を詳しく知ったので辛かった」って口コミに気になって。

ユーザ: 有名なところだよな！行ってみたいな。

システム: ね！

…(略)

図 6: 対話システムライブコンペ 2 予選での対話例。

- [6] Tatsuya Kawahara. Spoken dialogue system for a human-like conversational robot erica. In *International Workshop Spoken Dialogue Systems*, 2018.
- [7] Xu Li, Hyewon Lim, and Hyesun Hwang. Daily life with clova smart speaker: Topic modeling of clova review data. In *Proceedings of the XX International Conference on Human Computer Interaction*, p. 57. ACM, 2019.
- [8] Hiromi Narimatsu, Hiroaki Sugiyama, and Masahiro Mizukami. Detecting location-indicating phrases in user utterances for chat-oriented dialogue systems. In *LaCATODA@IJCAI*, pp. 8–13, 2018.
- [9] Ryuhei Nisimura, Yohei Nishihara, Ryosuke Tsurumi, Akinobu Lee, Hiroshi Saruwatari, and Kiyohiro Shikano. Takemaru-kun: Speech-oriented information system for real world research platform. 2003.
- [10] Oriol Vinyals and Quoc Le. A neural conversational model. *arXiv preprint arXiv:1506.05869*, 2015.
- [11] Yu Wu, Furu Wei, Shaohan Huang, Yunli Wang, Zhoujun Li, and Ming Zhou. Response generation by context-aware prototype editing. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 33, pp. 7281–7288, 2019.
- [12] 杉山弘晃, 成松宏美, 水上雅博, 有本庸浩. 文脈に沿った発話理解・生成を行うドメイン特化型雑談対話システムの実験的検討. Technical report, SIG-SLUD-B802-33, 人工知能学会, 2018.
- [13] 成松宏美, 杉山弘晃, 水上雅博, 有本庸浩, 宮崎昇. 文脈を理解して話す雑談対話システム. NTT 技術ジャーナル, p. 22, 2019.
- [14] 東中竜一郎, 船越孝太郎, 稲葉通将, 角森唯子, 高橋哲朗, 赤間怜奈, 宇佐美まゆみ, 川端良子, 水上雅博. 対話システムライブコンペティション 2. 人工知能学会 言語・音声理解と対話処理研究会第 87 回 (第 10 回対話システムシンポジウム), 2019.