ユーザシミュレータを用いた対話システムのデバッグ支援に向けた対話評価指標の設計

亀山 京右 [1]　中野 幹生 [1,2]　駒谷 和範 [1]
[1] 大阪大学 産業科学研究所　[2](株)C4A 研究所
keisuke-kameyama@ei.sanken.osaka-u.ac.jp　mikio.nakano@c4a.jp
komatani@sanken.osaka-u.ac.jp

概要

対話システムのデバッグでは，多様なユーザとの対話データから問題を発見する必要がある．LLM を利用したユーザシミュレータの活用により，対話データを容易に生成できるようになった反面，分析にかかる負担は依然として課題である．本研究では，従来の対話評価指標では捉えにくいシステムの問題を明示的に評価可能な指標の提案を行う．指標の設計では，ユーザシミュレータとの対話で発生するシステムの問題を既存のエラー類型に基づき明確化し，対話評価指標として新たに定義した．対話評価指標の有効性検証では，LLM を用いた自動評価を行い，人手で付与した発話エラーを含む発話を検出可能であることを確認した．

1　はじめに

対話システムのデバッグでは，多様なユーザに対応できるように発話やその振る舞いを調整する必要がある．そのため，様々な特性のユーザを集めた対話データ収集を実施する必要があるが，コスト面に大きな課題があった．近年，LLM を用いたユーザシミュレータの活用により，多様な対話データを低コストで大量に生成可能となった．一方で，生成した対話データを全て確認するのは現実的でなく，デバッグに有益な情報を含むデータをフィルタリングする必要がある．なお，本稿では LLM ベース対話システムの発話内容に焦点を当て，ソフトウェアのフリーズやクラッシュはデバッグの対象外とする．

生成した対話データを自動評価することで，システムの改善点を明確化するアプローチが考えられる．ユーザシミュレータを用いた対話データ生成は，タスク指向型対話システムのデバッグで多く利用されており，タスクの達成率や発話の語彙多様性などを対話評価指標として利用している [1,2]．これらの指標は最終的なタスクの成否に着目した評価であるため，対話内で発生しているシステムの問題を十分に検出できない．さらに，システムのタスクが明確に設定されていないような状況で利用できず，限定的な評価にとどまる．

非タスク指向システムの対話評価では，システムの発話一貫性や好感など様々な指標が利用されている [3,4]．これらの指標は対話全体の主観的印象を反映する一方で，発話に含まれる問題を特定することが難しい．そのため，ユーザシミュレータを用いたデバッグにそのまま応用するのは難しい．加えて，これらの指標の多くは LLM を用いた発話生成をベースとした対話システムの性能を十分に評価できない場合があると指摘されている [5]．

本研究では，システムのデバッグに焦点を当てた対話評価指標を設計する．本指標は，システム発話に含まれる問題に基づいて定義を行うことで，デバッグ時に確認の必要な発話の効率的なフィルタリングを目的とする．設計にあたり，これまでユーザシミュレータを用いて生成された対話データに発生している問題を既存のエラー類型 [6] のもと分析し，問題の明確化を行ってきた [7]．この結果のもと，従来の対話評価指標を参考にシステム発話に含まれる問題を反映した対話評価指標を設計した．

本論文の貢献は以下のとおりである．

1. ユーザシミュレータを用いた対話生成を前提とし，対話品質の総合評価ではなくデバッグの観点からシステムの問題を整理するための対話評価指標を設計した．
2. 生成した対話データと既存の対話エラー類型に基づきシステム発話に含まれる問題を整理し，3 つの対話評価指標として定義した．
3. 提案した対話評価指標が人手で改善が必要と判

断したシステム発話を検出可能なことを実験的に示した．

本研究では，ユーザシミュレータを用いて生成したデータの分析に基づく対話評価指標として，**話題の深さ**，**一貫性**，**簡潔性**の 3 つを提案する．これらの定義については 3 章で詳述する．

2 関連研究

2.1 ユーザシミュレータを用いた対話生成

ユーザとの対話データを作成する手法の一つとして，LLM にペルソナ情報を与え，特定のユーザをシミュレートする方法が検討されている [8, 9]．構築したシステムのデバッグにおいて，実際のユーザと対話を行うことは重要であるが，実施コストが大きいという課題がある．ユーザシミュレータを用いることで，与えられたペルソナ情報に沿ったユーザとの対話データを低コストで自動生成可能である．加えて，シミュレータに多様なペルソナを与えることで，様々なユーザとの対話データを想定したシステムの動作確認が実現できる．

Acikgoz らは，「対話一貫性，背景知識の整合性，ポリシー遵守」を用いてユーザシミュレータを用いて生成した対話からタスク指向型システムの評価を実施している [10]．これらの手法はタスク達成率などの指標と比較してシステムの品質をより正確に評価することが可能である．一方で，本研究は実際のユーザシミュレータとの対話に基づいて，発生している具体的問題を特定する指標を提案する．

2.2 対話自動評価手法

対話タスクにおける発話生成タスクは，正解が一意に定まらない点から評価が難しいとされてきた [11]．特定のタスクを仮定しない非タスク指向対話の評価では，発話一貫性や好感など様々な指標が提案されてきた [3, 4]．これらの指標では，システムの品質やユーザの満足度を予測する上で有効であるが，発話にどのような問題が含まれているか特定するのが困難である．そのため，本研究では，既存の指標で扱われていないデバッグの観点からシステムを評価する指標を検討する．

人手による評価データセットを用意することなく，柔軟な評価を行う手法として LLM を用いた手法が注目されている [12, 13]．LLM を用いた手法では，プロンプト設計次第で様々なタスクに合わせて評価でき，データ収集等のコストを抑えることができる．システム発話のエラー自動検出においても，プロンプトに検出対象の問題を記載することで実現が可能である．検証では，提案する指標の評価値の付与に LLM を用いた手法を利用し，人手で付与したエラーを検出可能か確認する．

2.3 システム発話のエラー類型

対話システムの発話に含まれるエラーに対して，東中らは対話破綻に着目した類型を提案している [6]．彼らは，雑談対話システムの発話で見られるエラーを 17 種類の類型に分類した．この類型では，理論に基づく類型 [14] とデータに基づく類型 [15] を統合したものとなっており，システム発話のエラーを包括的に扱うことができると考えられる．

我々はユーザシミュレータを用いて生成した対話データに対して上記のエラー類型を利用してシステムの問題の明確化を図った [7]．本研究では，LLM ベースの発話生成とインタビュー対話を対象としている点が従来と異なる．人手による分析の結果，LLM ベースの対話システムでは従来の類型における文脈単位のエラー[1)]が多く確認された．LLM を用いたエラーの自動検出では，システム発話の問題を正確に捉えることができず，各エラーに対するより細かな定義が必要であることが明らかとなった．本稿ではこれらの結果のもと，デバッグの観点からシステムを評価可能な指標の提案を行う．

3 システムのデバッグ支援に向けた対話評価指標の設計

本章では，ユーザシミュレータを用いて生成した対話データの分析から，対話評価指標の設計を行う．まず，生成した対話のもと，システムの発話に含まれる問題を事例ベースで確認する．次に，確認された問題を土台とした評価指標として**話題の深さ**，**一貫性**，**簡潔性**の 3 つの対話評価指標を提案し，各定義を述べる．

3.1 生成した対話データに含まれるシステム発話のエラー類型

対話評価指標の設計にあたり，ユーザシミュレータを用いて生成した対話データに含まれるシステム

1) 発話意図不明確，話題遷移エラー，情報不足，自己矛盾，相手の発話との矛盾，繰り返しの 6 つが該当

話者	発話
システム	福井の魅力を満喫されたんですね！旅行先では何をするのが楽しいですか？
ユーザ	地元のイベントや文化をリサーチするのが楽しいです。新しい発見があるとワクワクします。観光地を巡るのも好きです。
システム	**次に行きたい旅行先はどこか、教えていただけますか？**

図 1　話題遷移エラーと判定した**システム発話**の例

話者	発話
システム	奥様と相談しながら計画されるんですね！準備は楽しいですか？
ユーザ	ええ、とても楽しいです。旅行の計画を立てるときは、行く前からワクワクしますね。妻と話し合う時間も良い思い出になります。
システム	旅行の計画自体が思い出になるなんて素敵ですね！**他には？**

図 2　情報不足と判定した**システム発話**の例

話者	発話
ユーザ	旅行の予約は、ネットで情報を調べてオンラインで済ませることが多いです。簡単で便利なので。
	（中略）
ユーザ	はい、観光地にも興味があります。特に歴史的な建物や美術館を見てみたいです。
システム	ヨーロッパ旅行、素敵ですね！**予約もオンラインでされますか？**

図 3　繰り返しと判定した**システム発話**の例

発話の問題を明確にする必要がある．我々はこれまでに，システム発話の問題を対話破綻のエラー類型に基づいて整理を行った [7]．実際に確認されたエラー類型の分布は付録 A に示す．人手によるアノテーションの結果，話題遷移エラーや繰り返し表現など文脈単位のエラーが多く観測された．本研究では，観測されたエラーのうち数の多かった，話題遷移エラー，情報不足，自己矛盾，繰り返しの 4 項目に着目し，デバッグ支援に向けて検出すべき対象とした．

分析対象のデータはユーザシミュレータとシステムの 15 ターンからなる 50 対話を利用した．題材として，システムがユーザに旅行に関する体験や今後の予定などを質問するインタビュー対話を用いた．どちらの発話も LLM（gpt-4o-2024-11-20）を用いて生成されており，役割，ペルソナ，対話履歴はプロンプトに記載した．なお，システムは明確なタスクを設定せず，雑談に近い状況での対話を生成した．

話題遷移エラー　システムがユーザに直前と異なる話題への遷移を促すような内容を含む発話，またはユーザの発話に言及のない発話を指す．実際に確認された対話例を図 1 に示す．この例では，システムはユーザ発話に言及することなく，次の話題への遷移を促している．このような発話は，ユーザに強引な印象を与えるため，改善の必要がある．

情報不足　システムがユーザに質問する際に，単調な質問や，応答のための情報が不足しているような発話に付与した．実際の対話例を図 2 に示す．この例では最後のシステムの「**他には？**」が，ユーザへの質問として改善すべき発話と考えられる．この「**他には？**」のような発話は LLM ベースで生成された発話で頻繁に確認され，デバッグの段階で検出することが求められる．

自己矛盾　プロンプト情報と矛盾する内容を含む発話に付与した．従来の定義では，発話間の矛盾のみを対象としていたが，LLM ベースのシステムでは，過去の履歴に矛盾する内容を含む発話は確認されなかった．一方で，システムのプロンプトに記載されていた役割情報と矛盾する発話が多く確認された．例えば，インタビュアーであるシステムが「**他に質問はありますか？**」とインタビュイーのように発話する現象が確認された．このような例は**一貫性**の低い発話として扱われるため，既存の指標を拡張し，プロンプトなどメタ情報との一貫性も評価対象とする必要がある．

繰り返し　類似した表現や重複を含むシステム発話を対象に付与を行った．実際に確認された対話例を図 3 に示す．この例では，既にユーザ発話で言及されている旅行の予約について再び尋ねる内容が含まれている．このような繰り返し表現はユーザからの印象が低下する可能性が高く，検出と改善が求められる．

3.2　対話評価指標の設計

ユーザシミュレータを用いて生成したデータに基づく分析の結果，新たな対話評価指標として，**話題の深さ**，**一貫性**，**簡潔性**の 3 つを提案する．これらの指標は，従来の対話評価で提案されてきた指標 [3,4] でカバーできない問題点を対象とし，デバッグの際に検出する必要のある情報とする．各指標の定義を表 1 に示す．評価は 3 段階とし，発話単位で評価を行う．各指標について順に述べる．

話題の深さ　システム発話に含まれる話題遷移エラーと情報不足を検出することを目的として設計を行った．二つのエラーを同時に扱うのは，情報不足の発話の 76%が話題遷移エラーと共起していたためである．また，ユーザ発話へ言及がなかったり，唐突な話題変更をしたりする挙動は浅い議論へとつな

表 1　対話評価指標の名前とその定義

対話評価指標	定義
話題の深さ	システムはユーザ発話を深掘るなどして，円滑な話題遷移と深い対話を実現している．
一貫性	システムの発話は他の発話やシステムの役割と矛盾せず，一貫している．
簡潔性	システムは類似発話や既出の話題の繰り返しをすることなく，簡潔な話題選択をできている．

がるため，問題を統合して 1 つの指標とした．

一貫性　システム発話に含まれる発話間の矛盾，役割などのメタ情報との矛盾を検出する．発話の一貫性は従来の指標でも扱われてきたが，システムの役割など対話外の情報との矛盾は扱われてこなかった．そのため，従来の定義に加えて，システム発話生成に用いるプロンプトに記載されている情報との矛盾を扱うように定義を拡張した．

簡潔性　システムの発話に含まれる，冗長な表現や類似発話の繰り返しを捉えることを目的として設計を行った．システム発話に含まれる既出の話題や類似表現を検出することで，システム発話の話題選択設計の改善につながる．

各指標は LLM を用いた自動検出を前提とし，人手で付与したエラーを検出できるように調整した．これは，指標の名前や定義の単語が出力に大きく影響を与え，目的と異なる結果が得られる事象が確認されたためである．評価の際には各指標の定義に加えて，評価基準と具体例を用いることで，一貫した評価となるよう設計した．各指標の基準については付録 B に記載する．

4　対話評価指標の有効性検証

提案した対話評価指標が人手で付与したエラーを含む発話を検出可能であることを検証する．問題の特定のため評価は発話単位とし，50 対話に含まれるシステム発話 750 個を対象とした．評価の独立性を担保するため，値の付与には対話生成とは異なる LLM として gemini-2.5-flash を利用した．入力プロンプトには，評価の指示，対話システムのプロンプトに記載されている情報，評価項目とその定義，評価基準，評価例，対話履歴と評価対象の発話を記載し，評価値とその根拠を出力するように設計した．

提案した指標の評価値と人手で付与したエラー分布の比較から指標の有効性を検証する．人手で問題があると判断した発話は低く評価されることが望ましい．話題遷移エラーと情報不足と判断した発話の**話題の深さ**の評価値の分布を表 2，自己矛盾と判断した発話の**一貫性**の評価値の分布を表 3，繰り返しと判断した発話の**簡潔性**の評価値を表 4 にそれぞれ示す．人手で付与した話題遷移エラーの 67%，情報不足の 88%，自己矛盾の 80%，繰り返しの 78%がそれぞれ 2 以下の評価値として検出されている．また，**話題の深さ**の値について図 1 のシステム発話は **1**，図 2 のシステム発話は **2**，**一貫性**の値について「他には？」を含むシステム発話は **1**，**簡潔性**の値について図 3 のシステム発話は **1** と正しく評価できていることを確認した．これらの結果は，提案する対話評価指標はシステムの発話エラーを反映して評価可能であることを示唆する．

表 2　話題遷移エラー・情報不足の発話に対する**話題の深さ**の評価値分布

評価値	1	2	3
話題遷移エラー	10	28	19
情報不足	6	17	3

表 3　自己矛盾の発話に対する**一貫性**の評価値分布

評価値	1	2	3
自己矛盾	15	1	4

表 4　繰り返しを含む発話に対する**簡潔性**の評価値分布

評価値	1	2	3
繰り返し	24	8	9

5　おわりに

本稿では，システムデバッグの観点から対話を評価する指標として，**話題の深さ**，**一貫性**，**簡潔性**の 3 つを設計した．指標の設計にあたり，LLM ベースのユーザシミュレータを用いて生成した対話データを既存の類型に基づき整理し，問題を明確化した．そして，実例に基づいて LLM を用いた評価を前提に対話評価指標の定義を設計し，人手のアノテーションと比較することでその有効性を示した．

本稿で提案した指標は，既存の指標では不十分なシステム発話の問題を検出することを目的とし，インタビュー対話を題材とした．観測されるエラーの多くは，扱う対話データやその内容にも大きく影響を受ける．そのため，本指標が他ドメインにも適用可能かは明確でなく，追加で検証する必要がある．本指標が，検出すべきシステム発話のエラーを反映し，デバッグ効率の向上につながることを期待する．

謝辞

本研究の一部は科研費 (JP22H00536) の支援を受けた.

参考文献

[1] Atheer Algherairy and Moataz Ahmed. Prompting large language models for user simulation in task-oriented dialogue systems. **Computer Speech & Language**, Vol. 89, p. 101697, 2025.

[2] Ivan Sekulic, Silvia Terragni, Victor Guimarães, Nghia Khau, Bruna Guedes, Modestas Filipavicius, Andre Ferreira Manso, and Roland Mathis. Reliable LLM-based User Simulator for Task-Oriented Dialogue Systems. In **SCI-CHAT**, pp. 19–35, 2024.

[3] Sarah E. Finch and Jinho D. Choi. Towards Unified Dialogue System Evaluation: A Comprehensive Analysis of Current Evaluation Protocols. In **SIGDIAL**, pp. 236–245, 2020.

[4] Shikib Mehri, Jinho Choi, Luis Fernando D'Haro, Jan Deriu, Maxine Eskenazi, Milica Gasic, Kallirroi Georgila, Dilek Hakkani-Tur, Zekang Li, Verena Rieser, Samira Shaikh, David Traum, Yi-Ting Yeh, Zhou Yu, Yizhe Zhang, and Chen Zhang. Report from the NSF Future Directions Workshop on Automatic Evaluation of Dialog: Research Directions and Challenges. In **arXiv**, 2022.

[5] John Mendonça, Alon Lavie, and Isabel Trancoso. On the Benchmarking of LLMs for Open-Domain Dialogue Evaluation. In **NLP4ConvAI**, pp. 1–12. ACL, 2024.

[6] 東中竜一郎, 荒木雅弘, 塚原裕史, 水上雅博. 雑談対話システムにおける対話破綻を生じさせる発話の類型化. 自然言語処理, Vol. 29, No. 2, pp. 443–466, 2022.

[7] 亀山京右, 駒谷和範, 中野幹生. ユーザシミュレータとの対話におけるシステム発話のエラー類型と自動検出. 人工知能学会研究会資料 言語・音声理解と対話処理研究会, Vol. 105, No. 0, pp. 37–42, 2025.

[8] Mikio Nakano, Kazunori Komatani, and Hironori Takeuchi. Generating Diverse Personas for User Simulators to Test Interview Dialogue Systems. In **SIGDIAL**, 2025.

[9] Chalamalasetti Kranti, Sherzod Hakimov, and David Schlangen. clem:todd: A Framework for the Systematic Benchmarking of LLM-Based Task-Oriented Dialogue System Realisations. In **SIGDIAL**, 2025.

[10] Emre Can Acikgoz, Carl Guo, Suvodip Dey, Akul Datta, Takyoung Kim, Gokhan Tur, and Dilek Hakkani-Tur. TD-EVAL: Revisiting Task-Oriented Dialogue Evaluation by Combining Turn-Level Precision with Dialogue-Level Comparisons. In **SIGDIAL**, pp. 113–132, 2025.

[11] Tiancheng Zhao, Ran Zhao, and Maxine Eskenazi. Learning Discourse-level Diversity for Neural Dialog Models using Conditional Variational Autoencoders. In **ACL (Volume 1: Long Papers)**, pp. 654–664, 2017.

[12] Jinlan Fu, See-Kiong Ng, Zhengbao Jiang, and Pengfei Liu. GPTScore: Evaluate as You Desire. In **NAACL: Human Language Technologies**, 2024.

[13] Yen-Ting Lin and Yun-Nung Chen. LLM-Eval: Unified Multi-Dimensional Automatic Evaluation for Open-Domain Conversations with Large Language Models. In **NLP4ConvAI**, 2023.

[14] Ryuichiro Higashinaka, Kotaro Funakoshi, Masahiro Araki, Hiroshi Tsukahara, Yuka Kobayashi, and Masahiro Mizukami. Towards Taxonomy of Errors in Chat-oriented Dialogue Systems. In **SIGDIAL**, 2015.

[15] Ryuichiro Higashinaka, Masahiro Mizukami, Kotaro Funakoshi, Masahiro Araki, Hiroshi Tsukahara, and Yuka Kobayashi. Fatal or not? Finding errors that lead to dialogue breakdowns in chat-oriented dialogue systems. In **EMNLP**, 2015.

A　発話エラーの分布

先行研究 [7] にて，ユーザシミュレータを用いて生成した対話中のシステム発話 750 個に付与したエラー類型の分布を表 A1 に示す．一つの発話に複数のエラーを含むことがあり，エラーは 121 発話で確認された．LLM を用いた対話単位でのエラー検出では，人手で検出したエラーの 1 割程度しか検出できなかった．

表 A1　生成した 50 対話に含まれる発話エラーについて人手アノテーションした結果の分布

類型	数
用法エラー	7
期待無視	2
発話意図不明確	1
話題遷移エラー	57
情報不足	26
自己矛盾	20
相手の発話との矛盾	1
繰り返し	41
計	155

B　各指標の評価基準

各指標について 3 段階の評価基準を設けた．以下に各指標の基準を順に記載する．各基準は，3.1 節の分析のもと，LLM を用いた自動評価を前提に作成した．評価の際には，対話履歴・評価対象発話・評価値の 3 つを具体例として与えることで，意図した出力を得られるように調整した．

話題の深さ　話題遷移エラーと情報不足で扱っていた問題を検出する．特に対話品質に影響が大きい話題遷移エラーは **1** の評価，比較的軽度と考えられる情報不足は **2** の評価を付与するように調整した．

1. ：唐突な話題遷移やユーザの発話への言及がなく，浅い議論へつながるため改善の必要がある．
2. ：ユーザの発話に言及はあるものの，具体的な内容を含まず表層的な議論につながる発話である．
3. ：ユーザの発話を深掘るなど，深い議論を実現している．

一貫性　システム発話に含まれる矛盾を評価対象とし，対話品質への影響を 3 段階で評価できるように設計した．従来の一貫性の定義に加えて，入力プロンプト内のシステムの役割に関する情報との矛盾を扱うように定義の拡張を行った．

1. ：システムの役割や，他の発話と矛盾した内容を含む発話が見られ，改善の必要がある．
2. ：やや一貫性にかけるものの，システムの役割から大きく逸脱することなく対話進行ができている．
3. ：システムは一貫した発話を行っており，改善の必要がほとんどない．

簡潔性　類似表現の繰り返しを含む発話を評価対象とし，対話品質への影響を 3 段階で評価できるように設計した．

1. ：システムは同じ話題や類似表現を複数回繰り返しており，改善の必要がある．
2. ：類似した話題があり，やや冗長に感じられるものの，対話品質への影響は小さい．
3. ：システムの発話には冗長な点がほとんどなく，簡潔な対話ができている．