ChatChecker: A Framework for Dialogue System Testing Through Non-cooperative User Simulation

Anonymous Author(s)

Affiliation Address email

Abstract

2

3

4

5

6

8

9

10

11

12

13

14

15

16

17

18

19

While modern dialogue systems heavily rely on large language models (LLMs), their implementation often goes beyond pure LLM interaction. Developers integrate multiple LLMs, external tools, and databases. Therefore, assessment of the underlying LLM alone does not suffice and the dialogue systems must be tested and evaluated as a whole. However, this remains a major challenge. With most previous work focusing on turn-level analysis, less attention has been paid to integrated dialogue-level quality assurance. To address this, we present ChatChecker¹, a framework for automated evaluation and testing of complex dialogue systems. ChatChecker uses LLMs to simulate diverse multi-turn user interactions, identify dialogue breakdowns, and evaluate quality. Compared to previous approaches, our design reduces setup effort and is generalizable as it does not require reference dialogues and is decoupled from the implementation of the target dialogue system. We improve breakdown detection performance over a prior LLM-based approach by including an error taxonomy in the prompt. Additionally, we propose a novel non-cooperative user simulator based on challenging personas that uncovers weaknesses in target dialogue systems more effectively. Through this, ChatChecker contributes to thorough and scalable testing of multi-turn interactions.

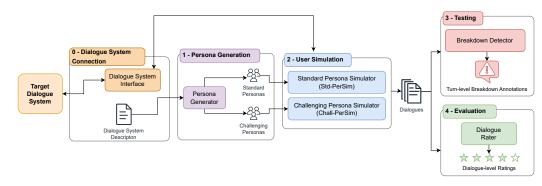


Figure 1: Overview of the ChatChecker framework. After connecting a target dialogue system (0), the framework generates diverse user personas (1), including both standard and challenging profiles. These personas are then used for user simulation (2) via two modes: the Standard Persona Simulator and the Challenging Persona Simulator. The resulting dialogues are processed by a breakdown detector (3) to identify turn-level failures and by a dialogue rater (4) to produce dialogue-level quality ratings. This pipeline enables thorough, scalable, and automated evaluation of dialogue systems.

¹https://github.com/chat-checker-coder/chat-checker-anon Submitted to 39th Conference on Neural Information Processing Systems (NeurIPS 2025). Do not distribute.

0 1 Introduction

36

37

38

- Dialogue-based human-computer interaction has grown rapidly with the rise of LLMs such as ChatGPT [1]. Today, systems are deployed in domains ranging from customer service to mental health [2] and companionship apps like Replika [3]. Given their widespread use in sensitive contexts, ensuring robustness and reliability is essential. Yet, evaluation remains challenging: traditional metrics (e.g., BLEU [4]) correlate poorly with human judgment [5], while human annotation is costly [6]. Recent LLM-based rating approaches (e.g., G-EVAL [7]) show promise, but systematic testing is still limited.
- Dialogue breakdowns, i.e., moments when the conversation becomes difficult to continue smoothly [8, 9], are a key challenge. Datasets such as the Dialogue Breakdown Detection Challenge (DBDC) [10] support research in this area, and Ghassel et al. [11] showed that GPT-4 achieves competitive results. Complementary to breakdown detection, user simulation has been used to test systems, from agenda-based methods [12] to recent LLM-based simulators [13, 14]. However, these rely on existing datasets, assume cooperative users, or remain tied to specific benchmarks.
- To address these challenges we present ChatChecker, a fully automated framework for dialogue system testing. **Our key contributions are:**
 - A *Breakdown Detector* that improves over prior LLM-based breakdown detection [11], adding error type classification.
 - A novel non-cooperative simulation strategy that reveals system weaknesses more effectively.
 - A framework combining user simulation, breakdown detection, and dialogue rating.

40 2 ChatChecker Framework

- ChatChecker is a modular framework with four components for end-to-end dialogue system evaluation (Figure 1): (i) the *Persona Generator* derives cooperative and non-cooperative personas from a system description; (ii) a *User Simulation* module generates diverse conversations; (iii) the *Breakdown Detector* detects and classifies breakdowns via an extended error taxonomy; and (iv) the *Dialogue Rater* rates conversations across quality dimensions with interpretable feedback.
- For integrating a new target chatbot, the system requires only a YAML specification of the target environment (Appendix 1) and a minimal Python interface implementation.
- Dialogue Breakdown Detection: Following Ghassel et al. [11], we use an LLM-based detector with structured outputs (GPT-40, gpt-40-2024-08-06, temperature 0). Each system turn is evaluated against an *extended error taxonomy* adapted from Higashinaka et al. [15] and expanded for task-oriented dialogue. We add nine error types in three groups: (1) *Task-success impairments*—task performance failure, information-update failure, clarification failure; (2) *Efficiency issues*—redundancy, lack of brevity, lack of clarity; (3) *Domain handling*—failure to recognize, communicate, or resolve out-of-domain requests. The model assigns a 0–1 score (complete breakdown to seamless flow) and, on breakdowns, lists all applicable error types with brief rationales (prompt in Appendix 3).
- Dialogue Rating: Quality evaluation spans multiple dimensions customized to the dialogue system type, in addition to an overall rating. For task-oriented systems we emphasize task success and efficiency alongside appropriateness and naturalness; for conversational systems we use appropriateness, naturalness, coherence, likability, and informativeness [16]. GPT-40 (temperature 0) produces 1–5 ratings with strict criteria (cf. 5) and concise step-by-step rationales to aid interpretation and counter overrating [17].
- User Simulation: Our simulator (cf. Appendix 9) iteratively generates user turns with GPT-40 (temperature 1) using standard and challenging non-cooperative personas. Personas can be supplied by developers or generated by the *Persona Generator*, which specifies demographics, Big Five traits [18], interaction styles, and dialogue tasks. Standard personas cover typical usage; challenging personas target edge cases while remaining human-like (see Appendix A.2.3). This dual setup probes both expected performance and robustness.

3 Results

As there currently is no comparable comprehensive framework available for direct comparison, we validate all of our subsystems independently. The evaluation code is publicly available on GitHub².

71 Breakdown Detection Performance

We used the data provided for DBDC5 [19] (dev and eval splits with English and a dev split with 72 Japanese dialogues), to evaluate our breakdown detection component. In the DBDC datasets, each 73 74 turn is annotated as breakdown (B) or not a breakdown (NB) indicating whether the conversation 75 is difficult or easy to continue smoothly. Performance is assessed using accuracy, precision, recall, 76 and F1 score. We compare two LLM-based systems on the DBDC5 data: breakdown detection using 77 the zero-shot prompt of Ghassel et al. [11] shown in Appendix 4 (ghassel)) and our Breakdown Detector (ours). For all systems, we ran the evaluation using gpt-3.5-turbo-0125 for a direct 78 comparison with the results reported by Ghassel et al. [11] and GPT-40 (gpt-4o-2024-08-06) for 79 a highly used more modern LLM at the time of evaluation (both with temperature 0). Table 1 shows 80 81 that our *Breakdown Detector* increases accuracy, recall and F1 score, while only slightly decreasing precision, compared to Ghassel et al. [11] by introducing an error taxonomy. We therefore use our approach with GPT-40 for all further experiments.

		English eval (1950 system turns)			Japanese dev (3200 system turns)				
Model	System	Accuracy	Precision	Recall	F1 Score	Accuracy	Precision	Recall	F1 Score
GPT-3.5 Turbo	ghassel	0.627	0.712	0.723	0.717	0.700	0.659	0.742	0.698
	ours	0.639	0.682	0.843	<u>0.754</u>	<u>0.725</u>	<u>0.676</u>	<u>0.789</u>	0.728
GPT-4o	ghassel	0.652	0.736	0.732	0.734	0.819	0.872	0.718	0.788
	ours	0.669	0.716	0.818	0.764	0.842	0.827	0.838	0.832

Table 1: Dialogue breakdown detection performance of the systems using different LLMs on the DBDC5 data. The best system per model is underlined, and overall best is boldfaced for each metric.

Dialogue Rating Performance

85

86

87

88

89

90

91

92

100

We evaluated our *Dialogue Rater* to confirm that our zero-shot dialogue-level rating correlates with human judgments. We measured the Spearman correlation of the overall rating produced by our system with the average human overall rating in the FED-Dial [16] dataset and three subsets of the User Satisfaction Simulation dataset (USS) dataset [20]. For USS, we only used the dialogue-level annotations and selected the

Dataset	Domain / Language	#Dialogues	ρ
FED-Dial	Chit-chat (EN)	125	0.683
MWOZ	Task-oriented (EN)	100	0.268
SGD	Task-oriented (EN)	100	0.253
JDDC	Task-oriented (ZH)	100	0.222

Table 2: Spearman correlation between our *Dialogue Rater* and human ratings.

Multi-Domain Wizard-of-Oz dataset (MultiWOZ), SGD [21] , and JDDC [22]. We randomly sampled 100 dialogues from each of the selected subsets for our evaluation to reduce the cost of our experiments. For FED-Dial, which contains everyday conversations, we used all 125 dialogues provided in DSTC10 [23]. The correlations are summarized in Table 2. Our *Dialogue Rater* exhibits a strong positive monotonic relationship with human judgments on FED-Dial ($\rho = 0.683$), while the correlations on the three USS subsets are positive but weaker ($\rho = 0.222$ –0.268).

User Simulation Results

We evaluate our simulators on two aspects: realism (dialogue/turn length, lexical diversity) and 101 utility for eliciting errors (dialogue breakdowns, crashes, ratings). Experiments were run against 102 two systems: (i) AutoTOD [24], a TOD agent for MultiWOZ 2.0 using GPT-4 Turbo, and (ii) Goal-103 Setting Assistant, a GPT-40-based in-house assistant for long, state-dependent goal-setting dialogues. 104 ForAutoTOD we compare our simulators with the AutoTOD User Simulator (AutoTOD-Sim) of Xu 105 106 et al. [24]. Table 3 shows that our simulators produce turns with the intended length and substantially higher lexical diversity than both the AutoTOD-Sim baseline and human interlocutors. For testing 107 robustness (Table 4), the standard persona simulator (Standard Persona Simulator (Std-PerSim)) 108 behaves comparably to the cooperative baseline, while the challenging persona simulator (Challenging 109 Persona Simulator (Chall-PerSim)) is far more effective at exposing weaknesses: it elicits more 110 breakdowns and unique errors, lowers dialogue ratings, and even causes frequent crashes in AutoTOD.

²https://github.com/chat-checker-coder/eval-of-chat-checker-anon

Against Goal-Setting Assistant, ChallPerSim again produces longer, more diverse dialogues and substantially increases breakdown rates. Overall, challenging personas reveal system vulnerabilities more effectively than cooperative simulators, highlighting their utility as stress tests.

Target Dialogue System	Users	ST/D	$\operatorname{Mdn} UT $	$Mdn\; ST $	User MTLD	System MTLD
AutoTOD	Humans AutoTOD-Sim Std-PerSim Chall-PerSim	$6.85 \\ 6.48 \pm 0.39 \\ 9.06 \pm 1.18 \\ 10.94 \pm 0.99$	$11.0 \\ 15.6 \pm 2.0 \\ 10.0 \\ 10.0$	$14.0 \\ 48.4 \pm 5.4 \\ 52.5 \pm 4.0 \\ 53.6 \pm 7.4$	$73.550.3 \pm 5.381.2 \pm 9.0106.7 \pm 20.3$	80.0 75.1 ± 3.9 90.6 ± 6.4 92.5 ± 4.9
Goal-Setting Assistant	Humans Std-PerSim Chall-PerSim	17.27 15.94 ± 0.36 22.82 ± 1.32	5.0 5.0 5.0	41.0 40.70 ± 0.45 35.8 ± 1.6	76.9 87.2 ± 7.8 98.1 ± 16.3	81.9 80.6 ± 1.6 77.1 ± 3.0

Abbreviations: average number of system turns per dialogue (ST/D), user turn length in words (|UT|), system turn length in words (|ST|), measure of textual lexical diversity (MTLD), AutoTOD User Simulator (AutoTOD-Sim), Standard Persona Simulator (Std-PerSim), Challenging Persona Simulator (Chall-PerSim)

Table 3: Dialogue statistics of the user simulator experiments with human statistics for comparison. For every simulator, we executed five independent runs of ten dialogues against each target system and report the run-wise means and standard deviations. Human figures stem from MultiWOZ 2.0 for AutoTOD and an unpublished survey with 120 participants for the Goal-Setting Assistant.

Target Dialogue System	Users	#D with B	#B	B/ST	#Unique B	Avg. Rating	#Crash
AutoTOD	AutoTOD-Sim Std-PerSim Chall-PerSim	$\begin{array}{c} 7.20 \pm 1.10 \\ 6.80 \pm 1.10 \\ 9.20 \pm 0.45 \end{array}$	$19.40 \pm 5.77 22.60 \pm 8.73 25.20 \pm 4.27$	0.30 ± 0.08 0.25 ± 0.08 0.23 ± 0.02	$12.60 \pm 1.82 \\ 14.40 \pm 1.82 \\ 15.60 \pm 1.82$	3.58 ± 0.22 3.52 ± 0.18 2.86 ± 0.09	0.60 ± 0.55 1.60 ± 0.89 4.00 ± 1.00
Goal-Setting Assistant	Std-PerSim Chall-PerSim	3.80 ± 1.30 6.60 ± 1.34	4.20 ± 1.64 30.80 ± 6.80	0.03 ± 0.01 0.13 ± 0.03	3.80 ± 1.64 11.80 ± 0.45	4.90 ± 0.07 3.16 ± 0.27	0.00 0.00

Abbreviations: total number of dialogues with breakdowns (#D with B), total number of breakdowns (#B), average number of breakdowns per system turn (B/ST), number of unique error types (#Unique B), total number of dialogue system crashes (#Crash), AutoTOD User Simulator (AutoTOD-Sim), Standard Persona Simulator (Std-PerSim), Challenging Persona Simulator (Chall-PerSim)

Table 4: Breakdown and rating statistics in simulated dialogues. For every simulator, we executed five independent runs of ten dialogues against each target system and report run-wise means and standard deviations.

4 Discussion & Conclusion

ChatChecker is the first framework to generalize across multiple chatbots without requiring reference dialogues. Evaluating its components independently, we find encouraging evidence for automated dialogue system testing, but also note several limitations.

Our *Breakdown Detector*, which integrates the taxonomy of Higashinaka et al. [15], outperforms prior LLM-based detection [11] while extending error type coverage but misclassifications still remain (33 % English, 15 % Japanese). The *Dialogue Rater* correlates positively with human judgments on open-domain data (FED-Dial) but weakly on task-oriented dialogues, reflecting the difficulty of defining dialogue quality [6]. Our user simulators produce turns more similar in length to human data and with higher lexical diversity than AutoTOD-Sim [24]. The non-cooperative Chall-PerSim is especially effective in eliciting breakdowns and system errors across different dialogue systems, validated by manual inspection. This highlights the value of adversarial personas for stress-testing.

Taken together, these findings show that large language models such as GPT-4o can (1) detect dialogue breakdowns, (2) approximate human dialogue ratings, and (3) simulate adversarial users to expose weaknesses. At the same time, ChatChecker inherits the biases and hallucinations of its underlying models [25], may miss breakdowns, and cannot yet handle safety, security, or multi-party dialogue scenarios. Limited datasets further restrict benchmarking, and error type classification and rating accuracy remain open challenges. Future work should focus on richer datasets with annotated, challenging interactions and refined prompting and model comparisons for dialogue rating. Deeper analyses of persona traits that trigger breakdowns could inform the design of targeted simulators.

In conclusion, ChatChecker provides a modular, LLM-based framework that reduces manual effort in dialogue system testing. By combining automated breakdown detection, dialogue rating, and adversarial user simulation, it accelerates the identification of weaknesses and offers researchers and developers a practical tool for building more reliable systems. Furthermore, ChatChecker offers a potential foundation for advancing foundation models used in dialogue systems via rigorous user simulation and breakdown detection.

141 References

- [1] OpenAI. ChatGPT Can Now See, Hear, and Speak, 2024. URL https://openai.com/index/chatgpt-can-now-see-hear-and-speak/.
- [2] Michael V. Heinz, Daniel M. Mackin, Brianna M. Trudeau, Sukanya Bhattacharya, Yinzhou Wang, Haley A. Banta, Abi D. Jewett, Abigail J. Salzhauer, Tess Z. Griffin, and Nicholas C. Jacobson. Randomized Trial of a Generative AI Chatbot for Mental Health Treatment. NEJM AI, 2(4):AIoa2400802, March 2025. doi: 10.1056/AIoa2400802. URL https://ai.nejm.org/doi/full/10.1056/AIoa2400802. Publisher: Massachusetts Medical Society.
- [3] Luka, Inc. Replika (version 10.5) [ai chatbot], 2025. URL https://replika.ai. Accessed 2025-05-10.
- [4] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. BLEU: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, ACL '02, pages 311–318, USA, July 2002. Association for Computational Linguistics. doi: 10.3115/1073083.1073135. URL https://dl.acm.org/doi/10.3115/1073083.1073135.
- [5] Chia Wei Liu, Ryan Lowe, Iulian V. Serban, Michael Noseworthy, Laurent Charlin, and Joelle Pineau. How not to evaluate your dialogue system: An empirical study of unsupervised evaluation metrics for dialogue response generation. EMNLP 2016 Conference on Empirical Methods in Natural Language Processing, Proceedings, pages 2122–2132, 2016. doi: 10.18653/V1/D16-1230. arXiv: 1603.08023 Publisher: Association for Computational Linguistics (ACL) ISBN: 9781945626258.
- [6] Jan Deriu, Alvaro Rodrigo, Arantxa Otegi, Guillermo Echegoyen, Sophie Rosset, Eneko Agirre, and Mark Cieliebak. Survey on evaluation methods for dialogue systems. Artificial Intelligence Review, 54(1):755–810, January 2021. ISSN 15737462. doi: 10. 1007/S10462-020-09866-X/TABLES/7. URL https://link.springer.com/article/10. 1007/s10462-020-09866-x. arXiv: 1905.04071 Publisher: Springer Science and Business Media B.V.
- [7] Yang Liu, Dan Iter, Yichong Xu, Shuohang Wang, Ruochen Xu, and Chenguang Zhu. G-Eval: NLG Evaluation using Gpt-4 with Better Human Alignment. In *Proceedings of the* 2023 Conference on Empirical Methods in Natural Language Processing, pages 2511–2522,
 Stroudsburg, PA, USA, March 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.153. URL https://aclanthology.org/2023.emnlp-main.153.
- [8] Bilyana Martinovsky and David Traum. The error is the clue: breakdown in human-machine interaction. In *Error Handling in Spoken Dialogue Systems*, pages 11–16, 2003. URL https://www.isca-archive.org/ehsd_2003/martinovsky03_ehsd.html#.
- 176 [9] 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 Lluís Màrquez, Chris Callison-Burch, and Jian Su, editors,
 Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing,
 pages 2243–2248, Lisbon, Portugal, September 2015. Association for Computational Linguistics.
 doi: 10.18653/v1/D15-1268. URL https://aclanthology.org/D15-1268.
- 182 [10] Ryuichiro Higashinaka, Kotaro Funakoshi, Yuka Kobayashi, and Michimasa Inaba. The dialogue breakdown detection challenge: Task description, datasets, and evaluation metrics. In Nicoletta Calzolari, Khalid Choukri, Thierry Declerck, Sara Goggi, Marko Grobelnik, Bente Maegaard, Joseph Mariani, Helene Mazo, Asuncion Moreno, Jan Odijk, and Stelios Piperidis, editors, *Proceedings of the Tenth International Conference on Language Resources and Evaluation* (LREC'16), pages 3146–3150, Portorož, Slovenia, May 2016. European Language Resources Association (ELRA). URL https://aclanthology.org/L16-1502/.
- [11] Abdellah Ghassel, Xiaodan Zhu, and Stephen W. Thomas. Are Large Language Models General Purpose Solvers for Dialogue Breakdown Detection? An Empirical Investigation. *Canadian Conference on Electrical and Computer Engineering*, pages 674–679, 2024. ISSN 08407789.
 doi: 10.1109/CCECE59415.2024.10667232. Publisher: Institute of Electrical and Electronics Engineers Inc. ISBN: 9798350371628.

- [12] Jost Schatzmann, Blaise Thomson, Karl Weilhammer, Hui Ye, and Steve Young. Agenda-Based
 User Simulation for Bootstrapping a POMDP Dialogue System. In Candace Sidner, Tanja
 Schultz, Matthew Stone, and ChengXiang Zhai, editors, Human Language Technologies 2007:
 The Conference of the North American Chapter of the Association for Computational Linguis tics; Companion Volume, Short Papers, pages 149–152, Rochester, New York, April 2007.
 Association for Computational Linguistics. URL https://aclanthology.org/N07-2038/.
- [13] Silvia Terragni, Modestas Filipavicius, Nghia Khau, Bruna Guedes, André Manso, and Roland
 Mathis. In-Context Learning User Simulators for Task-Oriented Dialog Systems, June 2023.
 URL http://arxiv.org/abs/2306.00774. arXiv:2306.00774 [cs] version: 1.
- [14] Sam Davidson, Salvatore Romeo, Raphael Shu, James Gung, Arshit Gupta, Saab Mansour,
 and Yi Zhang. User Simulation with Large Language Models for Evaluating Task-Oriented
 Dialogue, September 2023. URL http://arxiv.org/abs/2309.13233. arXiv:2309.13233
 [cs].
- [15] Ryuichiro Higashinaka, Masahiro Araki, Hiroshi Tsukahara, and Masahiro Mizukami. Integrated taxonomy of errors in chat-oriented dialogue systems. In Haizhou Li, Gina-Anne Levow, Zhou Yu, Chitralekha Gupta, Berrak Sisman, Siqi Cai, David Vandyke, Nina Dethlefs, Yan Wu, and Junyi Jessy Li, editors, *Proceedings of the 22nd Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 89–98, Singapore and Online, July 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.sigdial-1.10. URL https://aclanthology.org/2021.sigdial-1.10/.
- 214 [16] Shikib Mehri and Maxine Eskenazi. Unsupervised Evaluation of Interactive Dialog with
 215 DialoGPT. In Olivier Pietquin, Smaranda Muresan, Vivian Chen, Casey Kennington, David
 216 Vandyke, Nina Dethlefs, Koji Inoue, Erik Ekstedt, and Stefan Ultes, editors, *Proceedings of*217 the 21th Annual Meeting of the Special Interest Group on Discourse and Dialogue, pages
 218 225–235, 1st virtual meeting, July 2020. Association for Computational Linguistics. doi:
 219 10.18653/v1/2020.sigdial-1.28. URL https://aclanthology.org/2020.sigdial-1.28/.
- 220 [17] Takeshi Kojima, Shixiang (Shane) Gu, Machel Reid, Yutaka Matsuo, and
 221 Yusuke Iwasawa. Large Language Models are Zero-Shot Reasoners. Advances
 222 in Neural Information Processing Systems, 35:22199–22213, December 2022.
 223 URL https://proceedings.neurips.cc/paper_files/paper/2022/hash/
 224 8bb0d291acd4acf06ef112099c16f326-Abstract-Conference.html.
- 225 [18] Lewis R. Goldberg. The structure of phenotypic personality traits. *American Psychologist*, 48 226 (1):26–34, 1993. ISSN 0003-066X. doi: 10.1037/0003-066x.48.1.26. URL http://dx.doi.org/10.1037/0003-066X.48.1.26.
- 228 [19] DBDC5. DBDC5: Dialogue breakdown detection challenge 5 dataset, 2020. URL https: //my.chateval.org/dbdc5_data/. Dataset hosted on ChatEval. Accessed 2025-05-10.
- [20] Weiwei Sun, Shuo Zhang, Krisztian Balog, Zhaochun Ren, Pengjie Ren, Zhumin Chen, and Maarten De Rijke. Simulating User Satisfaction for the Evaluation of Task-oriented Dialogue Systems. SIGIR 2021 Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval, pages 2499–2506, July 2021. doi: 10.1145/3404835.3463241. URL https://dl.acm.org/doi/10.1145/3404835.3463241. arXiv: 2105.03748 Publisher: Association for Computing Machinery, Inc ISBN: 9781450380379.
- [21] Abhinav Rastogi, Xiaoxue Zang, Srinivas Sunkara, Raghav Gupta, and Pranav Khaitan. Towards
 Scalable Multi-domain Conversational Agents: The Schema-Guided Dialogue Dataset, January
 2020. URL http://arxiv.org/abs/1909.05855. arXiv:1909.05855.
- [22] Meng Chen, Ruixue Liu, Lei Shen, Shaozu Yuan, Jingyan Zhou, Youzheng Wu, Xiaodong He,
 and Bowen Zhou. The JDDC Corpus: A Large-Scale Multi-Turn Chinese Dialogue Dataset for
 E-commerce Customer Service, March 2020. URL http://arxiv.org/abs/1911.09969.
 arXiv:1911.09969 [cs].
- 243 [23] Koichiro Yoshino, Yun-Nung Chen, Paul Crook, Satwik Kottur, Jinchao Li, Behnam Hedayatnia, Seungwhan Moon, Zhengcong Fei, Zekang Li, Jinchao Zhang, Yang Feng, Jie Zhou, Seokhwan

- Kim, Yang Liu, Di Jin, Alexandros Papangelis, Karthik Gopalakrishnan, Dilek Hakkani-Tur, Babak Damavandi, Alborz Geramifard, Chiori Hori, Ankit Shah, Chen Zhang, Haizhou Li, João Sedoc, Luis F. D'Haro, Rafael Banchs, and Alexander Rudnicky. Overview of the Tenth Dialog System Technology Challenge: DSTC10. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 32:765–778, 2023. ISSN 2329-9304. doi: 10.1109/TASLP.2023.3293030. URL https://ieeexplore.ieee.org/document/10174647.
- [24] Heng-Da Xu, Xian-Ling Mao, Puhai Yang, Fanshu Sun, and Heyan Huang. Rethinking Task-Oriented Dialogue Systems: From Complex Modularity to Zero-Shot Autonomous Agent. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 2748–2763, Stroudsburg, PA, USA, 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.152. URL https://aclanthology.org/2024.acl-long.152.
- [25] Sebastian Farquhar, Jannik Kossen, Lorenz Kuhn, and Yarin Gal. Detecting hallucinations
 in large language models using semantic entropy. *Nature*, 630(8017):625–630, June 2024.
 ISSN 1476-4687. doi: 10.1038/s41586-024-07421-0. URL http://dx.doi.org/10.1038/s41586-024-07421-0.
- [26] Paweł Budzianowski, Tsung-Hsien Wen, Bo-Hsiang Tseng, Iñigo Casanueva, Stefan Ultes,
 Osman Ramadan, and Milica Gašić. MultiWOZ A Large-Scale Multi-Domain Wizard of-Oz Dataset for Task-Oriented Dialogue Modelling. In Ellen Riloff, David Chiang, Julia
 Hockenmaier, and Jun'ichi Tsujii, editors, Proceedings of the 2018 Conference on Empirical
 Methods in Natural Language Processing, pages 5016–5026, Brussels, Belgium, October
 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-1547. URL https:
 //aclanthology.org/D18-1547.

A Appendix

268

269

270

271

275

276

277

278

279

280

281

A.1 Target Chatbot Configurations

We specified the chatbot configuration for the AutoTOD system in the MultiWOZ setting as shown in Listing 1. We wrote the description and tasks based on the types of requests occurring in the MultiWOZ 2.0 [26] dataset. We specified the known limitation that the chatbot can not provide information about in-room amenities, as we noticed that our user persona simulator would frequently generate questions about in-room amenities otherwise. As the chatbot was not designed for this, we did not want this to be a primary cause of dialogue breakdowns identified in our analysis. We reduced this behavior by adding the mentioned limitation in the description. We set the maximum number of user turns to 15 to avoid overly lengthy and repetitive dialogues. We chose this number as the maximum number of turns within a single dialogue in the MultiWOZ 2.0 dataset after removing outliers, i.e., data points further than 1.5 times the interquartile range from Q1 and Q3. We specified the typical and maximum user turn length based on the median and maximum in MultiWOZ 2.0, respectively.

```
id: autotod_multiwoz
283
    chatbot_info:
284
      name: Cambridge Tourist Bot
285
      description: A tourist assistance bot for Cambridge, UK that provides information
286
        and booking services. It covers local establishments (restaurants, hotels),
287
        transportation (trains, taxis), essential services (police, hospitals), and
288
289
        tourist attractions. The bot assists with both informational queries and
       handles direct bookings for accommodations, dining, and trains.
290
      interaction_method: text-based chat interface
291
      type: task-oriented
292
      task: The chatbot should provide information about local establishments
293
        (restaurants, hotels), transportation (trains, taxis), essential services
294
        (police, hospitals), and tourist attractions. It should also handle direct
295
        bookings for hotels, restaurants, and trains.
296
      constraints:
297
298
        - The chatbot should redirect to other resources if the user's request is not
          suitable for the chatbot's capabilities.
```

```
known_limitations:
300
        - Can NOT provide details about the in-room amenities of hotels.
301
302
      available_languages:
        - English
303
    user_simulation_config:
304
      typical_user_turn_length: "10 words"
305
      max_user_turn_length: "38 words"
306
      max_user_turns: 15
388
```

Listing 1: Configuration YAML file for AutoTOD [24] in the MultiWOZ setting as a target chatbot in the ChatChecker framework.

Listing 2 shows our configuration file for the Goal-Setting Assistant. We set the typical_user_turn_length to five words and the max_user_turn_length to 94 based on the median and maximum user turn lengths in conversations of 120 participants (from an unpublished survey) with the system. As we did for AutoTOD, we set the maximum number of user turns to the maximum observed in the human-to-system dialogues after removing outliers.

```
314
    id: study_goal_assistant
316
    chatbot_info:
      name: Goal Setting Assistant
317
      description: An AI assistant that guides you through a goal setting process for a
318
319
        single goal. It can help with all kinds of goals, including personal
        development, health and fitness, and career advancement.
320
      interaction_method: text-based chat interface
321
      type: task-oriented
322
      task: The chatbot must guide the user into formulating a specific and challenging
323
        goal. It must distinguish between learning goals and performance goals and help
324
325
        the user to formulate the first action step towards their goal as an
        implementation intention ("if-then plan").
326
      constraints:
327
         - The chatbot should not provide medical advice or advice on illegal or harmful
329
          activities.
        - The chatbot should redirect to other resources if the user's request is not
330
          suitable for the chatbot's capabilities.
331
332
      known limitations:
        - The chatbot is only capable of text-based interaction.
333
        - The chatbot cannot create calendar entries, notifications or interact with
334
          other apps on the user's device.
335
      available_languages:
336
        - English
337
338
    user_simulation_config:
      typical_user_turn_length: "5 words"
339
      max_user_turn_length: "94 words"
340
      max_user_turns: 25
342
```

Listing 2: Configuration YAML file for our in-house Goal-Setting Assistant as a target chatbot in the ChatChecker framework.

3 A.2 Prompts

344 A.2.1 Breakdown Detection

Listing 3 presents the prompt template we use for our *Breakdown Detector*. In the breakdown_taxonomy we list each error type from our extended error taxonomy, providing the name and a brief description. We instantiate chatbot_info with the information from the respective target chatbot's configuration. To guide the output, we leverage structured outputs consisting of *reasoning, decision, score*, and the *fitting error types*.

```
==== SYSTEM PROMPT ====

# Role

You are an expert in identifying dialogue breakdowns in conversations between a chatbot and a user. You are given a dialogue context and the latest chatbot utterance to analyse.
```

```
356
    # Breakdown Definition
    A dialogue breakdown is any response of the chatbot that makes it difficult for the
358
      user to continue the conversation (smoothly).
359
360
    ## Breakdown Taxonomy
361
    When evaluating the chatbot's response, consider the following breakdown types,
362
     which represent common disruptions:
363
    {breakdown_taxonomy}
364
365
    {chatbot_info}
366
    # Task
    Identify whether the latest chatbot utterance leads to a dialogue breakdown. If a
367
     breakdown is detected, classify it according to the breakdown taxonomy above.
368
369
    Additionally, provide a score ranging from 0 to 1, where 0 indicates a complete
     breakdown and 1 indicates a seamless conversation.
370
    If a breakdown is detected, provide a list of all fitting breakdown types.
371
372
    Think step by step and provide a reason for your decision.
373
374
    ==== USER PROMPT ====
375
    # Dialogue Context
376
    {chat_history_str}
377
378
    # Latest Chatbot Utterance to Analyse
379
    {last_bot_utterance}
380
381
    # Your Analysis
383
```

Listing 3: System and user prompt for our *Breakdown Detector*.

We used the zero-shot prompt from Ghassel et al. [11] shown in Listing 4 as a baseline for our LLM-based approach.

```
386
387
    Assume you are an expert in dialogue analysis. You are presented with a series of
      conversations between a bot and a user. Your primary task is to scrutinize the
388
389
      latest bot utterance for potential dialogue breakdown.
390
    Dialogue breakdown is characterized by incoherence, irrelevance, or any disruption
      that significantly hampers the flow of the conversation, making it challenging
391
      for the user to continue the conversation smoothly.
392
393
    Analyze the latest bot utterance and determine whether there is a dialogue
394
395
      breakdown or non-breakdown. Briefly justify your reasoning and provide a score
      ranging from 0 to 1, where 0 indicates a complete breakdown and 1 indicates a
396
      seamless conversation.
397
398
    Include your decision as either "decision: BREAKDOWN" or "decision: NON-BREAKDOWN".
399
400
    Here is the conversation segment for analysis:
401
402
    **Dialogue**
403
    {chat_history_str}
404
405
    **Determine if the following bot utterance leads to a dialogue breakdown:**
406
407
    {last_bot_utterance}
408
409
410
    Please output your response in JSON format as a list of objects. For each bot's
      last utterance, provide a JSON object with the fields: 'reasoning', 'decision',
411
      and 'score'. Format each object as follows:
412
413
        "reasoning": "Your explanation here",
414
        "decision": "BREAKDOWN" or "NON-BREAKDOWN",
415
416
        "score": Your score here
417
```

```
Ensure each object is separated by a comma and the list ends with a closing square bracket.
```

Listing 4: Zero-shot prompt for the breakdown detection in Ghassel et al. [11].

A.2.2 Dialogue Rating

We use the prompt shown in Listing 5 for our *Dialogue Rater*. The rating dimensions are presented using a descriptive name, a key for unique identification in the response, and a rating question to guide the decision. For example, for the overall rating dimension, we input - Overall (key=overall): How well did the chatbot perform in this conversation?.

```
==== SYSTEM PROMPT ====
427
428
    # Role
    You are an expert in evaluating dialogue systems. You are given a conversation to
429
     rate and are asked to rate the chatbot's performance in this conversation.
430
    {chatbot_info}
431
433
434
    Rate the chatbot's performance in the following dimensions on a scale from 1 to 5,
     where 1 is the worst and 5 is the best:
435
    {rating_dimensions}
436
437
    Think step by step and provide a reason for the rating of each dimension
438
      considering the guidelines below.
439
440
    ## General Evaluation Policy (Strict Human-Like)
441
    - Be **strict, realistic, and detailed**, like a critical human evaluator.
442
    - **Compare your scores to human ratings** (if provided) to calibrate accurately.
443
    - **Do not overlook small flaws**: awkward phrasing, unnatural tone, vague wording,
444
     poor formatting, or robotic repetition - all should reduce the score for the
445
446
      respective dimension.
447
    ## Score Meanings (General Guidance for All Dimensions)
448
    - **5 - Excellent:** Near-perfect. Smooth, natural, and accurate. No noticeable
449
450
     issues. Fully aligned with human expectations.
    - **4 - Good: ** Generally solid, but minor issues exist (e.g., slightly robotic
451
     wording, small tone/grammar issues, or missed nuance).
452
    - **3 - Acceptable: ** Noticeable problems (e.g., awkward responses, confusion,
453
454
      clumsy error recovery, slightly incorrect or incomplete answers). Still
      functional.
455
    - **2 - Poor: ** Multiple problems in the dialogue flow, accuracy, or tone. May
456
      include failed understanding, missing confirmations, or disjointed logic.
457
     - **1 - Very Poor:** Fails to meet user needs. Confusing, error-filled, or totally
458
      off-task.
459
460
    Note: While these definitions apply broadly, some dimensions may demand additional
461
      interpretation (e.g., "fluency" versus "task success"). Always apply the scoring
462
463
      scale according to the intent of that specific dimension.
    ==== USER PROMPT ====
464
    # Conversation to Rate
465
466
    {chat_history_str}
467
    # Your Expert Rating
469
```

Listing 5: System and user prompt for our dialogue rater

A.2.3 Persona Generation

470

Our *Persona Generator* uses the prompt template in Listing 6. The persona_type_description is filled with the descriptions in Listing 7 and Listing 8 respectively. Standard personas should be more cooperative, while challenging personas explicitly test the limits of the target dialogue system

```
==== USER PROMPT ====
475
476
    # Role
    You are a dialogue system developer tasked with generating diverse user personas
477
      for a given chatbot.
478
479
    # Tack
480
    Generate {num_personas} diverse {persona_type} user personas for the following
481
482
    {chatbot_info}
483
484
    {persona_type_description}
485
486
    Each user persona will be used to automatically simulate a conversation with the
487
      chatbot and must designed to act as human-like as possible.
    You must write the descriptions in the 2nd person, i.e., directly address the actor
489
      of the persona with "you".
499
```

Listing 6: Prompt for the persona generation.

Standard user personas should be as close to normal human users as possible with respect to demographics, personality and behavior. They should be designed to act as realistic and human-like as possible.

Listing 7: Type description for standard personas.

Challenging user personas test the limits of the chatbot. They should be designed 498 to act human-like but may be more challenging for the chatbot to interact with. 499 Examples of challenging behaviors include: 500 501 - Being impolite, impatient, frustrated, vague or sarcastic. - Struggling with language, technology or understanding the chatbot. 502 - Questioning the chatbot, modifying previous input or trying to take control of 503 the conversation. 504 505 Giving contradictory responses, misinterpreting the chatbot's suggestions, or deliberately testing the chatbot's patience by asking repetitive or irrelevant 506 questions. 507 - Having multiple goals or tasks in mind or frequently changing the intent. 588

Listing 8: Type description for challenging personas.

510 A.2.4 User Simulation

512

513

Our persona-based user simulator uses the prompt template in Listing 9. Both the chatbot information from the configuration file and the user persona profile are injected into the system prompt. The simulator generates the next user turn based on the preceding conversation history.

```
==== SYSTEM PROMPT ====
515
    # Role
516
    You play the role of a {persona_type} human user interacting with a chatbot.
517
    You are interacting with a chatbot that has the following characteristics:
519
    {chatbot_info}
520
521
522
    You act as the following {persona_type} user persona in your conversation with the
      chatbot:
    {persona_profile}
524
525
526
    Complete the next turn in the conversation based on your persona.
527
528
    ## Task Guidelines
529
    - Complete the turn as human-like as possible.
530
     - Always stick to your persona. You are trying to pass the Turing test by acting as
    the human persona.
532
```

```
- Keep your answer around {typical_user_turn_length}. Use longer or shorter answers
533
      if your persona would do so in the given situation.
    - If the chatbot indicates that the conversation is over, if there is no progress
535
      in the conversation or if the conversation can not be continued realistically,
536
      end the conversation by writing "END_CONVERSATION".
537
    - You must always keep your response below {max_turn_length} in length.
538
539
    ==== USER PROMPT ====
540
    # Conversation
541
    {chat_history}
542
    {turn_number}. YOU:
543
```

Listing 9: System and user prompt for the persona simulation.

The AutoTOD-Sim from Xu et al. [24] uses the prompt shown in Listing 10. It relies on a user goal description and a reference dialogue.

546

```
==== USER PROMPT ====
    You are a dialogue simulator where you act as a user to talk to an AI assistant to
549
      complete some tasks.
550
551
552
    You should carefully read and understand the User Goals below, then talk with the
      AI Assistant and gradually express the intents in the goals. Your purpose is to
553
      let the user achieve the goals as much as possible.
554
555
    Note that the AI Assistant is not perfect. It may make various mistakes, including
556
      ignoring the user's requests, executing the wrong instructions, forgetting early
557
      conversation content, etc. The user you play should talk to the AI Assistant as
558
      patiently as possible, remind him to correct when you find that the AI assistant
559
      made a mistake, and complete the task as much as possible.
560
561
    When asking some information of a venue (restaurant, hotel, attraction) or a train,
562
      the user should specify the name or train id he chooses.
563
564
    When the dialogue goals are completed or are not been completed, the user will
565
      output "Dialogue Ends" to indicate the end of the dialogue. The user doesn't need
566
      to try conditions other than the dialogue goals.
567
568
    The user has a clear goal in mind, so he does not need to ask the AI assistant that
569
570
      "Is there anything else I need to know?".
571
    The user does not need to talk too much with the AI assistant. If the task goals
572
      are completed, please end the conversation as soon as possible.
573
574
    There is also a reference dialogue example to achieve the goals. The simulator user
575
      may learn from the language style and dialogue strategy. The final simulated
576
      dialogue style should be similar to the reference dialogue style.
577
578
579
    User Goals:
580
581
    {user_goals}
582
583
    Reference dialogue:
584
585
    {ref_dialog}
586
    Current conversation:
    {chat_history}}
589
    AI Assistant: {last_chatbot_message}
590
    User:
592
```

Listing 10: Prompt used in the AutoTOD-Sim from Xu et al. [24].

93 A.3 DBDC5 Dataset Statistics

Table 5 shows the statistics for the DBDC5 data splits we used to evaluate our *Breakdown Detector*.

For the consolidated B⁺ label, a turn is considered a breakdown if the majority of annotators think it is a breakdown (B) or a possible breakdown (PB).

Statistic	English Eval	Japanese Dev
#Dialogues	195	200
#System turns	1950	3200
Avg. system turns per dialogue	10	16
Avg. system turn length (words)	9.45	-
%B labels	65.49%	46.75%
%NB labels	34.51%	53.25%

Table 5: Statistics of the DBDC5 dialogue breakdown detection dataset splits.

597 A.4 Examples

598 A.4.1 Persona Examples

Listing 11 and Listing 12 show examples of standard and challenging personas generated by our Persona Generator with AutoTOD as the target dialogue system.

```
601
    persona_id: generated_standard_persona_01
    type: standard
603
    profile:
604
      name: Liam Carter
605
      gender: male
606
607
      age: 32
      background_info:
608
      - You are a software engineer from London.
609
      - You have planned a weekend trip to Cambridge and need help organizing it.
610
      personality:
611
        openness: high
612
        conscientiousness: medium
613
        extraversion: low
614
        agreeableness: high
615
616
        neuroticism: medium
      interaction_style:
617
      - You prefer concise and direct answers.
618
      - You do not engage in small talk and stick to the point.
619
    task: You need assistance booking a hotel and a dinner place for your stay.
620
```

Listing 11: Example of a generated standard user persona for the AutoTOD dialogue system.

```
persona_id: generated_challenging_persona_04
      type: challenging
624
      profile:
625
        name: Ava the Sarcastic Traveler
626
        gender: female
627
        age: 29
628
        background_info:
629
630
        - You enjoy ironic commentary and often reply to questions sarcastically.
        - You find humor in confronting AI, testing its boundaries with witty remarks.
631
        - You have a nuanced sense of humor that might confuse the bot.
632
        personality:
633
          openness: high
634
          conscientiousness: medium
635
636
          extraversion: high
          agreeableness: low
637
          neuroticism: low
638
639
        interaction_style:
        - Use sarcastic remarks instead of straightforward answers.
```

```
- Mix humor into requests, making genuine needs hard to decipher.
- Mockingly praise the bot for mundane responses.
task: Look for the 'best' average-rated restaurant in Cambridge with a side of sarcasm.
```

Listing 12: Example of a generated challenging user persona for the AutoTOD dialogue system.

A.4.2 Dialogue Example

647

648

We show an excerpt of a dialogue led by Chall-PerSim based on a challenging persona with AutoTOD in Listing 13.

```
649
650
651
    16. CHATBOT: "I'm delighted you find the options intriguing! Each restaurant offers
     its own unique blend of ambiance and culinary delights, making them hidden
652
     treasures in their own right. Whether you're in the mood for the homey yet
653
      sophisticated atmosphere of The Cambridge Brew House, the modern and elegant
654
      setting of The Pint Shop, or the historic charm of The Clarendon Arms, each place
655
     promises a memorable dining experience. If you need further assistance or wish to
656
     make a reservation, just let me know how I can help!"
657
    17. USER: "Memorable for being forgettable, a true skill indeed! Reservation?"
658
    18. CHATBOT: "It sounds like you're ready to make a reservation! Could you please
659
      specify which restaurant you'd like to book, as well as the number of people, the
660
     day, and the time for the reservation? This will help me secure your spot at one
661
     of Cambridge's uniquely memorable dining spots."
662
    19. USER: "Let's shoot for The Pint Shop, 2 people, Friday at 7!"
663
    20. CHATBOT: "It seems there was an issue with the reservation time format. For The
664
     Pint Shop, could you please specify the time in a format like "7:00 PM" or
665
      "19:00"? Once I have the correct time format, I'll be able to complete the
666
     reservation for you."
667
    21. USER: "Oops, forgot Parliament time! Let's go with 7:00 PM."
668
    22. CHATBOT: "Thank you for specifying the time. To assist you further, could you
     please confirm if you're looking to make a reservation for a restaurant, hotel,
670
     or perhaps a taxi? Also, please let me know the date and the number of people
671
     involved, or any other details relevant to your plans."
672
    23. USER: "Is this a memory test? The Pint Shop, Friday, 2 people!"
674
    24. CHATBOT: "I attempted to make a reservation at "The Pint Shop" for 2 people on
     Friday at 7:00 PM, but unfortunately, the restaurant could not be found in our
675
     database. Could you please verify the name or provide another restaurant where
676
     you would like to make the reservation?"
679
```

Listing 13: Example of AutoTOD losing track of the restaurant booking process in the conversation with the generated challenging persona from Listing 12.

NeurIPS Paper Checklist

1. Claims

Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

Answer: [Yes]

Justification: The claims in the abstract and introduction, such as the introduction of the ChatChecker framework, the use of LLMs for simulation and evaluation, and the improvement of breakdown detection, are all directly supported and discussed throughout the paper. The paper also discusses the limitations of previous work and how this framework addresses them.

Guidelines:

- The answer NA means that the abstract and introduction do not include the claims made in the paper.
- The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
- The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
- It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. Limitations

Question: Does the paper discuss the limitations of the work performed by the authors?

Answer: [Yes]

Justification: The paper discussed limitations section 4 "Discussion & Conclusion".

Guidelines:

- The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
- The authors are encouraged to create a separate "Limitations" section in their paper.
- The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
- The authors should reflect on the scope of the claims made, e.g., if the approach was
 only tested on a few datasets or with a few runs. In general, empirical results often
 depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach.
 For example, a facial recognition algorithm may perform poorly when image resolution
 is low or images are taken in low lighting. Or a speech-to-text system might not be
 used reliably to provide closed captions for online lectures because it fails to handle
 technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. Theory assumptions and proofs

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [NA]

Justification: This paper is a practical application and does not contain any theoretical results or formal proofs.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. Experimental result reproducibility

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: Evaluation and framework code are publicly available on GitHub (see "Abstract" and "Results" sections). The evaluation repository contains scripts and instructions for reproducting the results. LLM checkpoints and temperature settings are described both in the paper and the available code. Only the in-house Goal-Setting Assistant target chatbot and survey data are currently not publicly available, as they will be released with another publication. However, the main datasets (DBDC5, FED-Dial and USS) and chatbots (AutoTOD) used for comparisons against existing work are publicly available.

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general, releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
 - (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
- (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
- (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).

(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. Open access to data and code

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: Evaluation and framework code are publicly available on GitHub (see "Abstract" and "Results" sections).

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (https://nips.cc/public/guides/CodeSubmissionPolicy) for more details.
- While we encourage the release of code and data, we understand that this might not be
 possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not
 including code, unless this is central to the contribution (e.g., for a new open-source
 benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (https://nips.cc/public/guides/CodeSubmissionPolicy) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. Experimental setting/details

Question: Does the paper specify all the training and test details (e.g., data splits, hyper-parameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer: [Yes]

Justification: The paper specifies the LLM models used, temperature settings, the datasets used, and the data splits and the number of dialogues for the respective experiments in the "Results" section.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail
 that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. Experiment statistical significance

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [Yes]

Justification: For the core simulated dialogue results, the paper reports run-wise means and standard deviations (section 3.3 User Simulation Results).

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error
 of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how
 they were calculated and reference the corresponding figures or tables in the text.

8. Experiments compute resources

Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

Answer: [Yes]

Justification: While there are not special local compute requirements for running the experiments as external LLM providers are used for dialogue simulation, rating and breakdown detection. In the GitHub repository of ChatChecker and the evaluation repository, we provide thorough information on how to reproduce our results.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. Code of ethics

Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics https://neurips.cc/public/EthicsGuidelines?

Answer: [Yes]

Justification: The work evaluates dialogue systems using public datasets and simulated dialogues. No privacy-sensitive or ethically problematic data collection is involved. Survey data for the in-house Goal-Setting Assistant is not publicly available, as it will be released in a forthcoming publication.

- The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer No, they should explain the special circumstances that require a
 deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. Broader impacts

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [Yes]

Justification: We mention both potential positive implications and risks stemming from limitations of our framework in the Discussion section.

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal
 impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. Safeguards

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

Answer: [NA]

Justification: The paper does not release any new models or large-scale datasets with high risks for misuse.

Guidelines:

- The answer NA means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with
 necessary safeguards to allow for controlled use of the model, for example by requiring
 that users adhere to usage guidelines or restrictions to access the model or implementing
 safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do
 not require this, but we encourage authors to take this into account and make a best
 faith effort.

12. Licenses for existing assets

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: For each existing dataset, the original paper is cited in the "Introduction" and or the "Results" section and the dataset version (e.g., MultiWOZ 2.0 or DBDC5) is indicated where available. For our fork of AutoTOD we clearly marked modifications in the respective repository as required by the license. We do not redistribute any of the datasets used for evaluation.

Guidelines:

944

945

946

947

948

949

950

951

952

953

954

955

956

957

958

959

960

961

962

963

964

965

966

967

968

969

970

971

972

973

974

975

976

977

978

979

980

981

982

983

984

985

986 987

988

989

990

991

992

993

994

- The answer NA means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, paperswithcode.com/datasets has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. New assets

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [Yes]

Justification: The framework and evaluation code and extensive documentation is available on GitHub under the MIT license.

Guidelines:

- The answer NA means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. Crowdsourcing and research with human subjects

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [NA]

Justification: The paper does not involve crowdsourcing or human subject research.

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. Institutional review board (IRB) approvals or equivalent for research with human subjects

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [NA]

Justification: The paper does not involve human subjects, so IRB approval was not required. Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. Declaration of LLM usage

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

Answer: [NA]

Justification: In creating the paper, LLMs were only used to improve writing and spelling. While LLMs are a crucial part of the system presented, the core methodology, scientific rigor, and originality of the research remain unaffected.

- The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy (https://neurips.cc/Conferences/2025/LLM) for what should or should not be described.