MASTER'S THESIS

Automated Exploration and Profiling of Conversational Agents

Master's in Data Science

Author:	Iván Sotillo del Horno
Supervisor:	Juan de Lara Jaramillo
Co-supervisor:	Esther Guerra Sánchez
Department:	Department of Computer Science
Submission Date:	July 11, 2025

Universidad Autónoma de Madrid
Escuela Politécnica Superior

I confirm that this master's is my own work and I have documented all sources and material used.

Madrid, Spain, July 11, 2025 Iván Sotillo del Horno

Acknowledgments

Abstract

Contents

1 Introduction

The proliferation of conversational agents, commonly refered to as chatbots, has fundamentally transformed the landscape of human-computer interaction accross diverse domains. From general-purpose assistants such as OpenAI's ChatGPT [1] or Google's Gemini [2] to task-oriented agents that assist users in specific tasks like shopping or customer support. These systems allow for natural language interaction with services ranging from customer support and e-commerce platforms to educational resources. The proliferation of these agents has been further accelerated by advances in generative Artificial Intelligence (AI), particularly Large Language Models (LLMs), which have significantly enhanced chatbot capabilities, allowing them to both create and understand natural language without explicitly programmed rules.

The presence of these agents in so many applications has elevated concerns regarding their reliability, correctness, and quality assurance. As these systems appear in domains such as healthcare or finances, which require high levels of trust, the need for rigorous testing and validation becomes paramount. However, the heterogeneous nature of chatbot development, with intent-based frameworks like Google's Dialogflow [3] or Rasa [4], multi-agent programming environments built upon LLMs such as LangGraph [5] and Microsoft's AutoGen [6], and Domain-Specific Languages (DSLs) like Taskyto [7], presents significant challenges for finding a comprehensive methodology to test these systems.

Traditional software testing techniques are very limited when applied to chatbot systems. The complexity of Natural Language Processing (NLP), the non-deterministic nature of LLMs and the dynamic flow of a real conversation make traditional testing inadequate for conversational agents. While there have been some approaches for developing testing techniques for chatbots [8, 9], they often target specific chatbot technologies [10], require substantial manual effort including the provision of test conversations [10, 11] or synchronous human interaction [12], depend on existing conversation corpus [13], or need access to the chatbot's source code [14–16], thereby limiting their applicability to deployed systems treated as black boxes.

The research presented in this thesis aims to solve these problems through the development of Task Recognition And Chatbot ExploreR (TRACER), a tool for extracting comprehensive models from deployed conversational agents, and then, with this model, create user profiles which serve as test cases for a user simulator called Sensei [17].

TRACER employs an LLM agent to systematically explore the chatbot's capabilities via natural language interactions, eliminating the need for manual test case creation or access to the chatbot's source code. This black-box approach enables the automated generation of detailed chatbot models that encapsulate supported languages, fallback mechanisms, functional capabilities, input parameters, admissible parameter values, output data structures, and conversational flow patterns.

The extracted chatbot model serves as the foundation for the automated synthesis of test cases. Specifically, TRACER generates user profiles that represent diverse users that interact with the chatbot using Sensei [17], but different implementations of TRACER could be made to generate different types of test cases based on the extracted model. The integration of TRACER with Sensei yields a testing methodology that only requires a connector for the chatbot's API.

To ensure the accessibility and reproducibility of this research, TRACER has been developed as a complete, open-source tool. It is publicly available as a Python Package Index (PyPI) package [18] and can be installed via `pip install chatbot-tracer`. The full source code is hosted on GitHub https://github.com/Chatbot-TRACER/TRACER, and a dedicated web application has been developed to provide a user-friendly experience for the entire testing pipeline, from model extraction and user profiles generation with TRACER to test execution with Sensei.

To guide this investigation, we have defined the following research questions:

- **RQ1: How effective is TRACER in modeling chatbot functionality?** This question assesses the ability of our model exploration technique to achieve high functional coverage in a controlled setting where the ground truth is known.

- **RQ2: How effective are the synthesized profiles at detecting faults in controlled environments?** This question evaluates the precision of our approach by using mutation testing [15] to measure the ability of the generated profiles to identify specific, injected faults.

- **RQ3: How effective is the approach at identifying real-world bugs and ensuring task completion in deployed chatbots?** This addresses the practical, real-world applicability of our framework by measuring the Bug Detection Rate (BDR) and Task Completion Rate (TCR) of the generated profiles against real-world chatbots.

Thesis structure. Chapter 2 establishes the background and state of the art in chatbot test. Chapter 3 presents the core methodoly of how TRACER extracts models from chatbots. Chapter 4 describes the structure of the user profiles, and how TRACER generates them. Chapter 5 shows TRACER Command Line Interface (CLI) and the web application to use both TRACER and Sensei. Chapter 6 presents the evaluation of TRACER against the research questions. Chapter 7 concludes the thesis and discusses future work.

2 Background and State of the Art

2.1 Background

2.1.1 Conversational Agents

Conversational agents, commonly referred to as chatbots, are software systems designed to interact with users through natural language dialogue. These systems have evolved from simple rule-based programs that followed predefined conversation flows to sophisticated AI-powered agents capable of understanding context, maintaining conversational state, and generating human-like responses.

Modern conversational agents can be categorized into two main types given the domain and range of their capabilities.

- **Task-oriented:** on one hand we have the task-oriented chatbots, these are designed to assist users in completing specific tasks, such as booking appointments, processing orders, or providing customer support. These systems typically follow structured conversation flows and maintain explicit state management to track task progress. Examples of these chatbots are Taskyto [7] or UAM's assistant Ada [19].

- **Open-domain:** on the other hand, open-domain chatbots engage in general conversation without specific task constraints, aiming to provide informative, helpful, or entertaining interactions across a wide range of topics. These are chatbots like ChatGPT [1] or Gemini [2].

The development of these conversational agents has beeen facilitated by various frameworks and platforms.

- **Intent-based frameworks:** these frameworks such as Google's Dialogflow [3] or Rasa [4] enable developers to define conversation flow through intents, utterances, and responses. These platforms have low latency and deterministic behavior but are very rigid, struggle to scale, and to work properly require a big corpus to be trained on.

- **Multi-agent programming environments:** these systems like LangGraph [5] or Microsoft's AutoGen [6], allow for the creating of complex conversational systems where multiple AI agents collaborate to process the user's request. These frameworks make use of the capabilities of LLMs. While they are less rigid than the previous ones, they can suffer from hallucinations, higher latency, and since they are not deterministic, getting out of the scope, and thus, making it harder to test it.

2.1.2 Large Language Models

Large Language Models represent a significant advancement in Natural Language Processing, enable conversational agents to understand and generate human-like text without explit programming of conversational rules like in intent-based frameworks. These models, trained on a vast ammount of text data, have demonstrated remarkable capabilities in language understanding [20], generation, and reasoning across diverse domains.

The integration of LLMs into conversational agents has transformed the way humans interacts with computers. Unlike traditional rules-based systems that rely on predefined patterns and responses, LLM-powered chatbots can engage in natural conversations, even keeping context about what the user said before. However, this flexibility comes with challenges, specially for testing and validation. The non-deterministic natura of LLMs means that identical inputs may produce different outputs across multiple interactions, making traditional testing approaches inadequate for testing and validating them.

The just mentioned behavior exhibited by LLM-powered systems further complicates testing efforts. These systems can demonstrate capabilities that were not explicitly programmed, making it difficult to predict all possible conversation paths and outcomes. This unpredictability necessitates new approaches to testing that can systematically explore the space of possible interactions and validate system behavior across diverse scenarios.

2.1.3 Black-box Testing

Black-box testing is a software testing methodology where the internal structure, implementation details, and source code of the system under test are unknown or inaccessible to the tester. This approach focuses on validating system behavior based solely on inputs and outputs, treating the system as an opaque "black box."

In the context of conversational agents, black-box testing presents an interesting opportunity. The accessibility advantage of black-box testing is particularly relevant for deployed chatbots, where users and testers typically interact with systems through APIs or web interfaces without access to underlying code or configuration. This mirrors real-world usage scenarios and enables testing of production systems without requiring special access privileges or development environment setup.

However, black-box testing of conversational agents faces unique challenges. The exploration problem involves systematically discovering the full range of functionalities and conversation paths supported by the chatbot. Unlike traditional software systems with well-defined APIs, conversational agents accept natural language input, making the input space virtually infinite. The validation challenge requires determining whether chatbot responses are correct, appropriate, and helpful without access to specifications or expected behavior definitions.

3 TRACER: Automated Chatbot Exploration

4 User Profile Structure and Generation

5 Tool Support

6 Evaluation

7 Conclusions and Future Work

Abbreviations

AI Artificial Intelligence

NLP Natural Language Processing

LLM Large Language Model

DSL Domain-Specific Language

TRACER Task Recognition And Chatbot ExploreR

BDR Bug Detection Rate

TCR Task Completion Rate

CLI Command Line Interface

PyPI Python Package Index

List of Figures

List of Tables

Bibliography

[1] "ChatGPT," Accessed: Jul. 10, 2025. [Online]. Available: https://chatgpt.com.

[2] "Google Gemini," Gemini, Accessed: Jul. 10, 2025. [Online]. Available: https://gemini.google.com.

[3] "Dialogflow," Google Cloud, Accessed: Jul. 10, 2025. [Online]. Available: https://cloud.google.com/products/conversational-agents.

[4] "Rasa," Rasa, Accessed: Jul. 10, 2025. [Online]. Available: https://rasa.com/.

[5] "LangGraph," Accessed: Jul. 10, 2025. [Online]. Available: https://www.langchain.com/langgraph.

[6] "AutoGen," Accessed: Jul. 10, 2025. [Online]. Available: https://microsoft.github.io/autogen/stable/.

[7] J. Sánchez Cuadrado, S. Pérez-Soler, E. Guerra, and J. De Lara, "Automating the Development of Task-oriented LLM-based Chatbots," in *Proceedings of the 6th ACM Conference on Conversational User Interfaces*, ser. CUI '24, New York, NY, USA: Association for Computing Machinery, Jul. 8, 2024, pp. 1–10, ISBN: 979-8-4007-0511-3. DOI: 10.1145/3640794.3665538. Accessed: Mar. 19, 2025. [Online]. Available: https://doi.org/10.1145/3640794.3665538.

[8] J. S. Cuadrado, D. Ávila, S. Pérez-Soler, P. C. Cañizares, E. Guerra, and J. De Lara, "Integrating Static Quality Assurance in CI Chatbot Development Workflows," *IEEE Software*, vol. 41, no. 5, pp. 60–69, Sep. 2024, ISSN: 0740-7459, 1937-4194. DOI: 10.1109/ms.2024.3401551. Accessed: Jul. 10, 2025. [Online]. Available: https://ieeexplore.ieee.org/document/10533225/.

[9] P. C. Cañizares, J. M. López-Morales, S. Pérez-Soler, E. Guerra, and J. De Lara, "Measuring and Clustering Heterogeneous Chatbot Designs," *ACM Transactions on Software Engineering and Methodology*, vol. 33, no. 4, pp. 1–43, May 31, 2024, ISSN: 1049-331X, 1557-7392. DOI: 10.1145/3637228. Accessed: Jul. 10, 2025. [Online]. Available: https://dl.acm.org/doi/10.1145/3637228.

[10] "Rasa Test," Accessed: Jul. 10, 2025. [Online]. Available: https://rasa.com/docs/pro/testing/evaluating-assistant/.

[11] "Cyara Botium," Cyara, Accessed: Jul. 10, 2025. [Online]. Available: `https://cyara.com/products/botium/`.

[12] R. Ren, J. W. Castro, S. T. Acuña, and J. De Lara, "Evaluation Techniques for Chatbot Usability: A Systematic Mapping Study," *International Journal of Software Engineering and Knowledge Engineering*, vol. 29, pp. 1673–1702, 11n12 Nov. 2019, ISSN: 0218-1940, 1793-6403. DOI: `10.1142/s0218194019400163`. Accessed: Jul. 10, 2025. [Online]. Available: `https://www.worldscientific.com/doi/abs/10.1142/S0218194019400163`.

[13] M. Vasconcelos, H. Candello, C. Pinhanez, and T. Dos Santos, "Bottester: Testing Conversational Systems with Simulated Users," in *Proceedings of the XVI Brazilian Symposium on Human Factors in Computing Systems*, Joinville Brazil: ACM, Oct. 23, 2017, pp. 1–4. DOI: `10.1145/3160504.3160584`. Accessed: Jul. 10, 2025. [Online]. Available: `https://dl.acm.org/doi/10.1145/3160504.3160584`.

[14] P. C. Cañizares, D. Ávila, S. Perez-Soler, E. Guerra, and J. de Lara, "Coverage-based Strategies for the Automated Synthesis of Test Scenarios for Conversational Agents," in *Proceedings of the 5th ACM/IEEE International Conference on Automation of Software Test* (*AST 2024*), ser. AST '24, New York, NY, USA: Association for Computing Machinery, Jun. 10, 2024, pp. 23–33, ISBN: 979-8-4007-0588-5. DOI: `10.1145/3644032.3644456`. Accessed: Mar. 19, 2025. [Online]. Available: `https://doi.org/10.1145/3644032.3644456`.

[15] P. Gómez-Abajo, S. Pérez-Soler, P. C. Cañizares, E. Guerra, and J. de Lara, "Mutation Testing for Task-Oriented Chatbots," in *Proceedings of the 28th International Conference on Evaluation and Assessment in Software Engineering*, ser. EASE '24, New York, NY, USA: Association for Computing Machinery, Jun. 18, 2024, pp. 232–241, ISBN: 979-8-4007-1701-7. DOI: `10.1145/3661167.3661220`. Accessed: Mar. 19, 2025. [Online]. Available: `https://doi.org/10.1145/3661167.3661220`.

[16] M. F. Urrico, D. Clerissi, and L. Mariani, "MutaBot: A Mutation Testing Approach for Chatbots," in *Proceedings of the 2024 IEEE/ACM 46th International Conference on Software Engineering: Companion Proceedings*, Lisbon Portugal: ACM, Apr. 14, 2024, pp. 79–83. DOI: `10.1145/3639478.3640032`. Accessed: Jul. 10, 2025. [Online]. Available: `https://dl.acm.org/doi/10.1145/3639478.3640032`.

[17] J. De Lara, E. Guerra, A. del Pozzo, and J. Sanchez Cuadrado. "Sensei," GitHub, Accessed: Jul. 10, 2025. [Online]. Available: `https://github.com/satori-chatbots/user-simulator`.

[18] I. Sotillo del Horno, *Chatbot-tracer: A tool to model chatbots and create profiles to test them.* Version 0.2.10. Accessed: Jul. 10, 2025. [Online]. Available: `https://github.com/Chatbot-TRACER/TRACER`.

[19] "AdaUAM," Accessed: Jul. 11, 2025. [Online]. Available: `https://www.uam.es/uam/tecnologias-informacion/servicios-ti/acceso-remoto-red`.

[20] S. Li, Y. Chen, and X. Zhang, "Enhancing Natural Language Instruction Document Comprehension with Large Language Models," in *2024 5th International Conference on Computer, Big Data and Artificial Intelligence (ICCBD+AI)*, Jingdezhen, China: IEEE, Nov. 1, 2024, pp. 622–626. DOI: `10.1109/iccbd-ai65562.2024.00109`. Accessed: Jul. 11, 2025. [Online]. Available: `https://ieeexplore.ieee.org/document/10933784/`.