

Assessing Task-based Chatbots: Snapshot and Curated Datasets for Dialogflow

Elena Masserini

University of Milano Bicocca
Milan, Italy
elena.masserini@unimib.it

Daniela Micucci

University of Milano Bicocca
Milan, Italy
daniela.micucci@unimib.it

Diego Clerissi

University of Milano Bicocca
Milan, Italy
diego.clerissi@unimib.it

Leonardo Mariani

University of Milano Bicocca
Milan, Italy
leonardo.mariani@unimib.it

Abstract

In recent years, chatbots have gained widespread adoption thanks to their ability to assist users at any time and across diverse domains. However, the lack of large-scale curated datasets limits research on their quality and reliability. This paper presents TOFU-D, a snapshot of 1,788 Dialogflow chatbots from GitHub, and COD, a curated subset of TOFU-D including 185 validated chatbots. The two datasets capture a wide range of domains, languages, and implementation patterns, offering a sound basis for empirical studies on chatbot quality and security. A preliminary assessment using the Botium testing framework and the Bandit static analyzer revealed gaps in test coverage and frequent security vulnerabilities in several chatbots, highlighting the need for systematic, multi-platform research on chatbot quality and security.

CCS Concepts

• Software and its engineering;

Keywords

Chatbot, Dialogflow, Dataset, GitHub

ACM Reference Format:

Elena Masserini, Diego Clerissi, Daniela Micucci, and Leonardo Mariani. 2026. Assessing Task-based Chatbots: Snapshot and Curated Datasets for Dialogflow. In *23rd International Conference on Mining Software Repositories (MSR '26)*, April 13–14, 2026, Rio de Janeiro, Brazil. ACM, New York, NY, USA, 5 pages. <https://doi.org/10.1145/3793302.3793325>

1 Introduction

Task-based chatbots are ubiquitous in a multitude of domains [2], such as healthcare, bookings, and personal assistance. Different from *general-purpose* chatbots (e.g., ChatGPT [21]) that do not usually target a specific domain and are not yet able to deliver mature and reliable services, task-based chatbots can fulfill domain-specific tasks [3, 28] and represent consolidated solutions extensively used in industry [2, 9]. Task-based chatbots can be implemented with

both commercial (e.g., Dialogflow [24] and Amazon Lex [32]) and open-source (e.g., Rasa [39]) platforms. All these platforms use the same set of core concepts about conversations: the *intents*, which represent the goals users may want to achieve by interacting with chatbots (e.g., ordering pizza); the *utterances*, which are the phrases used to train the chatbot to understand user intents; the *entities*, which are the data types used in the conversations; the *actions*, which are the responses and operations that the chatbot can do to address the user intents; and the *flows*, which implement the possible user-bot sequence of interactions [20, 26].

The popularity and complexity of chatbots, which combine conversational data with backend logic, demand proper quality assurance techniques. In fact, traditional quality assurance techniques do not adapt straightforwardly to NLP-based software [17, 31, 33]. Botium [11] is a multi-platform test generation and execution framework for task-based chatbots. The framework has been exploited as a backbone in multiple studies on test case augmentation [16, 38], scenario generation [18], and mutation testing [22, 26, 27]. Related studies also considered black-box testing [13, 23, 43], the oracle problem [12, 14], runtime verification [41], quality metrics [19], robustness [29, 30, 34, 40], and security concerns [10, 15].

A common issue with the proposed approaches is the small-scale empirical evidence that is often reported, where subjects are selected without specific criteria, resulting in the consideration of a few and simple chatbots only. The lack of proper datasets is one of the main factors contributing to this issue. For instance, the dataset released with ASYMOB [19, 36] consists of a collection of chatbots selected in the wild without using any specific criteria, and many of them are outdated nowadays. In a recent work [37], we presented TOFU-R, a snapshot of the Rasa chatbots present on GitHub, and BRASATO, a curated selection of 193 Rasa chatbots. Although useful, these datasets include chatbots developed with the same open-source platform.

This paper addresses these limitations by presenting two new datasets with chatbots implemented for Dialogflow [24], the popular commercial Google platform for the development of task-based chatbots [1]. The first dataset is TOFU-D (DIALOGFLOW TASK-BASED CHATBOTS FROM GITHUB), containing a snapshot of all the Dialogflow chatbots present on Github on September 16th 2025, (1,788 chatbots). The second dataset is COD (COLLECTION OF DIALOGFLOW CHATBOTS), a curated selection of the chatbots present in TOFU-D (185 chatbots). The two new datasets enable *stronger empirical*



This work is licensed under a Creative Commons Attribution 4.0 International License. *MSR '26, Rio de Janeiro, Brazil*

© 2026 Copyright held by the owner/author(s).
ACM ISBN 979-8-4007-2474-9/26/04
<https://doi.org/10.1145/3793302.3793325>

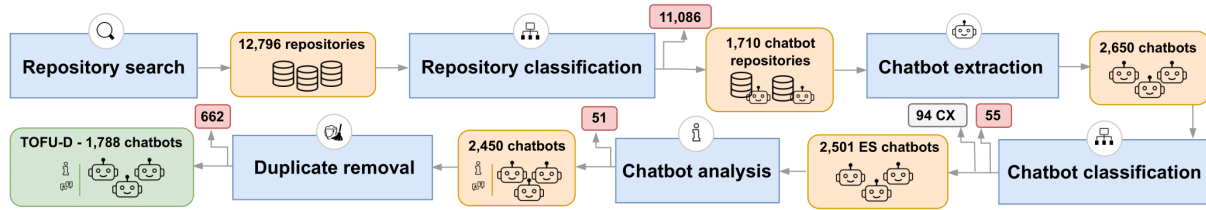


Figure 1: Methodology for the creation of the TOFU-D dataset.

research in the area according to multiple perspectives: (a) they improve *generality*, since they include chatbots developed for a commercial platform that have different architecture and characteristics (e.g., different domains, different conversational capabilities, and different complexity) than chatbots for open-source platforms, as reported in Section 3, (b) they increase *programming language heterogeneity*, since Dialogflow chatbots are implemented with a variety of languages while Rasa chatbots are all implemented in Python, and (c) they increase the *diversity* and *number* of the chatbots available, by doubling the number of curated chatbots available.

The main contributions of the paper are the TOFU-D and COD datasets for Dialogflow and the toolchain designed to create them. TOFU-D represents the state-of-the-practice of Dialogflow chatbots on GitHub. COD enables quality assurance research on experiments larger than prior studies, which rarely considered more than 10 chatbots [27, 37]. The toolchain enables replicability and allows adaptation to other research goals. Further, we report a preliminary experiment using testing and security tools with a selection of chatbots, which serves as an illustrative example to demonstrate the practical utility of COD, confirming the challenges already observed with other datasets, and also revealing new specific challenges.

2 Methodology

Here we discuss the methodology adopted to create the TOFU-D and the COD datasets for Dialogflow. The methodology is *fully automated* and can be generalized to other task-based frameworks [20].

2.1 Extraction of the TOFU-D dataset

Figure 1 shows the methodology used to create the TOFU-D dataset.

Repository search: We identified publicly available GitHub repositories that may include Dialogflow chatbots. We used the GitHub repository API [5] to collect all the repositories that contain the keyword "Dialogflow" and at least one between "chatbot" and "agent" in the name, README file, description, or list of topics. This search resulted in 12,883 repositories. To obtain a permanent reference to the chatbot version available on Sept. 16th 2025, we stored the latest commit SHA of each repository. This process revealed 87 empty repositories that we discarded, resulting in 12,796 non-empty ones.

Repository classification: The occurrence of the searched keywords in a repository does not ensure the presence of an actual chatbot, since other repositories may also include those terms (e.g., repositories with chatbot development frameworks or testing tools). To confirm the presence of chatbots in the repositories, we downloaded the repositories and checked for the presence of the *agent file*, which

is a configuration file required by Dialogflow chatbots. This step excluded 11,086 repositories without chatbots, resulting in 2,650 agent files in 1,710 repositories.

Chatbot extraction: Similarly to previous studies [37], we assume that distinct agent files in distinct folders indicate distinct agents. Since all agent files within the same repository were in distinct folders, we confirmed the availability of the 2,650 Dialogflow chatbots.

Chatbot classification: Google Dialogflow chatbots can be designed to run either on Dialogflow Essentials (ES), a free platform intended for individual developers, or Dialogflow Customer Experience (CX), the platform for enterprise-scale systems. In this work, we targeted Dialogflow chatbots that can be executed without requiring a subscription. The analysis of the agent files revealed the presence of 94 CX chatbots and 2,501 ES chatbots, while 55 chatbots were discarded due to an incorrect agent file structure.

Chatbot analysis: To provide a comprehensive description of the chatbots, we extracted the key elements defining their structure and behavior. Structural information includes the list of handled *intents*, the recognized *entities*, a brief *description* (when available in the agent file), and the *Dialogflow version*. Behavioral information includes the set of *supported conversational languages*, the presence of an integration with Google Assistant [7], and any backend code implementing *webhook services*, which are the services that are executed when a more complex operation than a plain response is needed. We also collected information on the *webhook intents*, which are the intents that actually trigger the execution of webhook services, and the webhook services that are implemented through cloud functions. In this step, we discarded 51 chatbots due to an incorrect structure of the implementation.

Duplicate removal: To avoid duplicates in the final TOFU-D dataset (e.g., due to the same chatbot code uploaded to different repositories), we checked for the presence of *nearly identical chatbots*, which are chatbots that share the same list of intents, entities, webhook intents, and supported languages, and have a *very similar* webhook service implementation. Specifically, we treated two chatbots as distinct only if their backend code similarity, assessed by the `diff1ib` [25] library for sequence comparison, is below 95%, thus accounting for minor differences, such as code comments. For each group of duplicates, we selected the most representative chatbot, prioritizing the most recent Dialogflow version, to favor up-to-date subjects, higher popularity (in terms of stars and forks), and, in case of ties, the older creation date representing the original chatbot. This process led to the removal of 662 copies, resulting in the final TOFU-D dataset of 1,788 unique Dialogflow chatbots.

2.2 Chatbot selection for the COD dataset

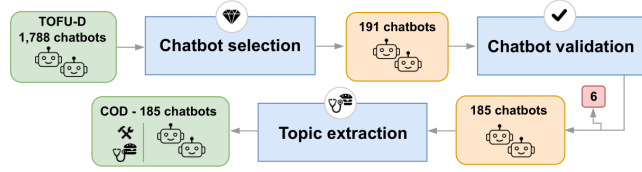


Figure 2: Methodology for the creation of the COD dataset.

While TOFU-D provides a comprehensive overview of the Dialogflow chatbot population on GitHub, deriving a curated dataset with the most *relevant chatbots* is important to facilitate future studies. We define as relevant the chatbots selected from TOFU-D based on three criteria that capture challenging and relevant aspects for quality assurance research [37]: a minimal *dialog complexity*, which requires that the chatbot can handle at least one intent and one entity; *functional complexity*, which requires the chatbot to use a webhook service, available in the repository; and *utility*, which requires the chatbot to support English (to be widely usable), to support the latest Dialogflow version (v2), and to be stored in a starred repository. As summarized in Figure 2, only 191 chatbots out of 1,788 satisfy these criteria. Each chatbot was then validated through the Dialogflow REST API [4] to ensure successful deployment, which led to the removal of six chatbots due to entity configuration issues. Lastly, to identify the topics that are typically addressed with chatbots, we designed a prompt that, for each chatbot, takes its repository title and description, its list of intents and entities, and its README files, asking GPT-4o to associate each chatbot with a Google Play Category [6] to minimize variability. To finally validate the correctness of the pipeline, an author performed a sample-based inspection of each intermediate result (e.g., to validate the topic accuracy provided by GPT-4o), not detecting any misclassification.

3 Datasets Analysis

Dialog complexity. The boxplots in Figure 3 (a) show the dialog complexity of TOFU-D and COD, measured as the number of implemented intents and entities (using logarithmic scale). Both boxplots highlight a spread distribution in TOFU-D, with the number of intents and entities ranging from 1 to 316 and from 0 to 140, respectively. In particular, TOFU-D includes several chatbots (15%) that do not process any information. This subset consists of both toy examples and chatbots that uniquely implement long enumerations of predefined answers to predefined user queries. COD discards these cases, as well as chatbots without any backend implementation, resulting in a more curated selection of chatbots providing actual services through conversations. Note that trivial chatbots responding according to long pre-defined enumerations may include many intents and entities; this is why COD does not include some apparently large, but irrelevant, chatbots.

Functional complexity. This higher complexity of COD, on the functional side, is highlighted by the analysis of intent types in the two datasets, as shown in Figure 3 (b). All chatbots in COD implement at least one action different from the mere output of a predefined

textual response (i.e., have at least one webhook intent). Moreover, the majority of the chatbots (62%) exploit the webhook service for more than half of the possible user intents.

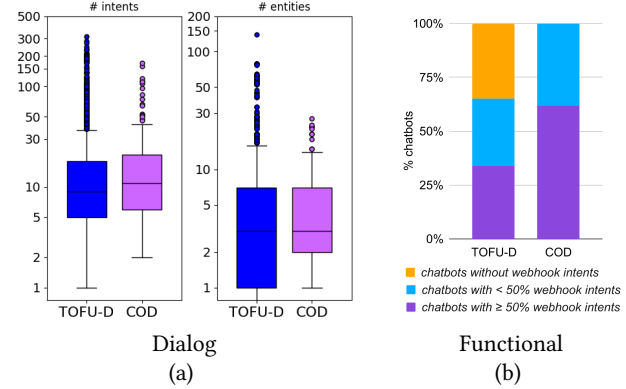


Figure 3: Complexity of TOFU-D and COD.

An interesting aspect of Dialogflow chatbots is their possible integration with Google’s proprietary services, such as the Google Assistant and Google cloud functions. These two capabilities are relatively present in TOFU-D population, with only 32% and 16% of chatbots supporting them, respectively. Their frequency is definitely higher in COD, where 51% of the chatbots support Google Assistant and 29% of them implement cloud functions. This result confirms COD is representative of modern chatbots that integrate with cloud features and state-of-the-art assistants.

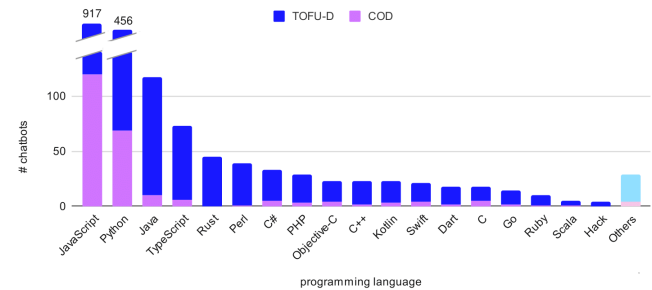


Figure 4: Programming languages in TOFU-D and COD.

Figure 4 plots the number of chatbots that include backend files in the most popular programming languages. Since chatbots are systems that must dynamically and flexibly react to user requests, developers favor programming languages with strong support to an event-driven design, like JavaScript, Python, Java, and TypeScript. JavaScript is particularly popular since it is natively supported by Dialogflow and the Google development platform.

Utility. Lastly, we discuss utility in terms of supported languages, Dialogflow version, and popularity of the repositories, with a final insight into the chatbot domains in COD. As shown in Figure 5, the language distribution in TOFU-D reveals a strong predominance of English, supported by 1,557 chatbots. The set of the most popular languages is mainly consistent with the set of the most spoken languages in 2025 [42], though with some differences in content (e.g., Italian is not one of the most spoken languages) and ranking (e.g., Chinese being less represented in TOFU-D). Although the

standard is a monolingual implementation, a total of 159 chatbots support multiple languages, with up to 12 languages supported in three chatbots. Another common standard is the use of the latest platform version: 82% of the chatbots in TOFU-D rely on Dialogflow v2, due to its availability since 2019 and the early deprecation of the previous one [35]. Regarding popularity, only 37% of the chatbots in TOFU-D come from starred projects on GitHub; while the number of stars for the repositories that originate the chatbots in COD ranges from 1 to 289. Chatbots in COD cover 28 distinct Google Play domains, with Education (18%), Business (13%), Medical (11%), and Food and Drinks (7%) being the most common topics.

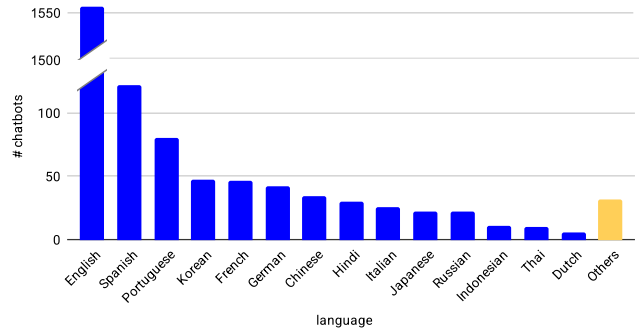


Figure 5: Conversational languages in TOFU-D.

4 Early Validation with COD

The datasets described in this work can support and inspire future research on task-based chatbot development and quality. TOFU-D and COD can be used to explore how correct, efficient, robust, and secure task-based chatbots are, with particular emphasis on software developed by the open-source community. Moreover, our datasets, COD in particular, can be used to explore the effectiveness of state-of-the-art quality assurance methods, to finally identify the research gaps that must be addressed in the future.

Below, we describe a preliminary investigation we conducted using COD about (a) the effectiveness of *automatic testing* techniques, and (b) the *vulnerabilities* that may affect task-based chatbots.

To explore the effectiveness of test case generation tools, we randomly selected 10 chatbots from COD¹ and generated test cases with the state-of-the-art tool Botium [11], a widely used framework for the functional assessment of task-based chatbots. We observed *missing tests for intents* managing fallback behaviors (100% of chatbots affected), greeting the user (30%), or requiring preconditions from previous interactions to activate (30%). We also observed cases of completely *uncovered entities* (30% of chatbots affected), or only partially covered ones (70%). Furthermore, we observed cases of *broken tests*, where the chatbot response was missing (10%). Considering the results reported for Rasa chatbots [37], these preliminary findings confirm that designing effective approaches for automatic test generation is a general cross-platform challenge.

To explore how vulnerable chatbots are, we used Bandit, a static analysis tool [8] designed to detect security vulnerabilities in Python code. We analyzed all the 69 chatbots in COD that include Python

¹The selected chatbots are: *BestCovid19_bot-DialogFlow*, *ASD-Bot*, *PyTacos*, *Smart-School-Chatbot*, *Dr.Bot*, *Driving-Pal*, *ChatterBot-TextClassifier*, *dbpedia-chatbot-backend*, *echo*, and *conversational-ai-history-agent*.

code to perform a comparison with all the 193 chatbots in BRASATO, which are always implemented in Python.

Table 1 shows the six most common security issues detected in both datasets and the corresponding share of affected chatbots. The higher number of chatbots affected by potential vulnerabilities in COD is likely due to their higher complexity compared to the chatbots in BRASATO, which usually have a smaller backend.

Interestingly, the results show commonalities and differences between the two classes of chatbots. On one hand, missing timeouts in external APIs requests, the use of weak pseudo-random generators, improper handling of exceptions (e.g., try-catch-pass) and possible SQL injections caused by unsanitized input parameters are common to chatbots implemented with both platforms, suggesting that chatbot developers may easily overlook the integration of external APIs, databases, and random generators when implementing chatbots. On the other hand, Dialogflow chatbots exhibit vulnerabilities specific to their webhook-based architectures, including API misconfigurations that accept connections from any network interface (41% of the chatbots) and that may allow the execution of arbitrary code (32% of the chatbots).

These results show how task-based chatbots must be studied according to a multi-platform perspective, which allows researchers to address both the general challenges that characterize the chatbot technology and the ones that are specific to certain platforms.

Table 1: Most common security issues in BRASATO and COD.

BRASATO		COD	
Issue	Freq.	Issue	Freq.
request without timeout	29%	all interfaces binding	41%
pseudo-random generator	17%	flask debug true	32%
pickle deserialization	5%	request without timeout	32%
hardcoded SQL string	5%	pesudo-random generator	26%
silent exception handling	5%	silent exception handling	15%
assert ignored by compiler	2%	hardcoded SQL string	10%

5 Conclusions

This paper presents TOFU-D, a snapshot of the Dialogflow chatbots present on GitHub, and COD, a curated dataset of 185 Dialogflow chatbots. These datasets represent a step towards a more systematic quality assessment of chatbots, as they provide a benchmark of real-world subjects for the validation and development of novel strategies. In particular, COD: i) provides a collection of chatbots implemented for a different (commercial) platform compared to Rasa [37], mitigating the risk of developing research that overfits a single technology, ii) provides chatbots implemented in multiple programming languages, iii) doubles the total number of curated chatbots available (185 in COD, 193 in BRASATO) for experimentation, iv) includes chatbots with unique features, such as cloud functions and Google Assistant technology, v) includes chatbots with testing and security challenges complementary to existing datasets. Future work concerns the use of the datasets to systematically develop and experiment with quality assurance methods.

Data Availability. The datasets and the scripts are publicly available at <https://gitlab.com/securitychatbot/dialogflow-chatbot-dataset>.

References

- [1] Ahmad Abdellatif, Khaled Badran, Diego Elias Costa, and Emad Shihab. 2021. A comparison of natural language understanding platforms for chatbots in software engineering. *IEEE Transactions on Software Engineering (TSE)* 48, 8 (2021), 3087–3102.
- [2] Eleni Adamopoulou and Lefteris Moussiades. 2020. Chatbots: History, technology, and applications. *Machine Learning with Applications* 2 (2020), 100006.
- [3] Eleni Adamopoulou and Lefteris Moussiades. 2020. An overview of chatbot technology. In *Proceedings of the International Conference on Artificial Intelligence Applications and Innovations (ALAI)*.
- [4] Dialogflow REST API. Accessed January 2026. *Dialogflow API Documentation*. <https://docs.cloud.google.com/dialogflow/es/docs/reference/rest/v2-overview>
- [5] GitHub Repository API. Accessed January 2026. *REST API endpoints for repositories*. <https://docs.github.com/en/rest/repos>
- [6] Google Search API. Accessed January 2026. *Google Play Store Categories*. <https://www.searchapi.io/docs/parameters/google-play-store/categories>
- [7] Google Assistant. Accessed January 2026. *Google Assistant Official Website*. <https://assistant.google.com/>
- [8] Bandit. Accessed January 2026. *Bandit Documentation*. <https://bandit.readthedocs.io/en/latest/>
- [9] Lamy Benaddi, Charaf Ouaddi, Abdeslam Jakimi, and Brahim Ouchao. 2024. A systematic review of chatbots: Classification, development, and their impact on tourism. *IEEE Access* 12 (2024), 78799–78810.
- [10] Domna Bilika, Nikoleta Michopoulou, Efthimios Alepis, and Constantinos Patsakis. 2024. Hello me, meet the real me: Voice synthesis attacks on voice assistants. *Computers & Security* 137 (2024), 103617.
- [11] Botium. Accessed January 2026. *Botium Documentation*. <https://botium-docs.readthedocs.io/en/latest>
- [12] Josip Božić. 2022. Ontology-based metamorphic testing for chatbots. *Software Quality Journal (SQJ)* 30, 1 (2022), 227–251.
- [13] Josip Bozic, Oliver A Tazl, and Franz Wotawa. 2019. Chatbot testing using AI planning. In *Proceedings of the International Conference On Artificial Intelligence Testing (AITest)*.
- [14] Josip Bozic and Franz Wotawa. 2019. Testing chatbots using metamorphic relations. In *Proceedings of the International Conference on Testing Software and Systems (ICTSS)*.
- [15] Josip Bozic and Franz Wotawa. 2020. Interrogating Virtual Agents: In Quest of Security Vulnerabilities. In *Proceedings of the International Conference on Testing Software and Systems (ICTSS)*.
- [16] Sergio Bravo-Santos, Esther Guerra, and Juan de Lara. 2020. Testing chatbots with Charm. In *Proceedings of the International Conference on the Quality of Information and Communications Technology (QUATIC)*.
- [17] Jordi Cabot, Loli Burgueño, Robert Clarisó, Gwendal Daniel, Jorge Perianez-Pascual, and Roberto Rodríguez-Echeverría. 2021. Testing challenges for NLP-intensive bots. In *Proceedings of the International Workshop on Bots in Software Engineering (BotSE)*.
- [18] Pablo C Cañizares, Daniel Ávila, Sara Pérez-Soler, Esther Guerra, and Juan de Lara. 2024. Coverage-based Strategies for the Automated Synthesis of Test Scenarios for Conversational Agents. In *Proceedings of the International Conference on Automation of Software Test (AST)*.
- [19] Pablo C Cañizares, Jose María López-Morales, Sara Pérez-Soler, Esther Guerra, and Juan de Lara. 2024. Measuring and clustering heterogeneous chatbot designs. *ACM Transactions on Software Engineering and Methodology (TOSEM)* 33, 4 (2024), 1–43.
- [20] Pablo C Cañizares, Sara Pérez-Soler, Esther Guerra, and Juan de Lara. 2022. Automating the measurement of heterogeneous chatbot designs. In *Proceedings of the Symposium on Applied Computing (SAC)*.
- [21] ChatGPT. Accessed January 2026. *ChatGPT Web Interface*. <https://chatgpt.com/>
- [22] Diego Clerissi, Elena Masserini, Daniela Micucci, and Leonardo Mariani. 2025. Towards Multi-Platform Mutation Testing of Task-based Chatbots. In *Proceedings of the 9th International Workshop on Software Faults (IWSF)*.
- [23] Iván Sotillo del Horno, Alejandro del Pozzo, Esther Guerra, and Juan de Lara. 2025. Automated Exploration of Conversational Agents for the Synthesis of Testing Profiles. In *IFIP International Conference on Testing Software and Systems (ICTSS)*. Springer, 213–230.
- [24] Dialogflow. Accessed January 2026. *Dialogflow Documentation*. <https://docs.cloud.google.com/dialogflow/docs>
- [25] difflib. Accessed January 2026. *Python difflib Documentation*. <https://docs.python.org/3/library/difflib.html>
- [26] Michael Ferdinando Urrico, Diego Clerissi, and Leonardo Mariani. 2024. MutaBot: A Mutation Testing Approach for Chatbots. In *Proceedings of the International Conference on Software Engineering, Companion (ICSE-C)*.
- [27] Pablo Gómez-Abajo, Sara Pérez-Soler, Pablo C Cañizares, Esther Guerra, and Juan de Lara. 2024. Mutation Testing for Task-Oriented Chatbots. In *Proceedings of the International Conference on Evaluation and Assessment in Software Engineering (EASE)*.
- [28] Jonathan Grudin and Richard Jacques. 2019. Chatbots, humbots, and the quest for artificial general intelligence. In *Proceedings of the Conference on Human Factors in Computing Systems (CHI)*.
- [29] Jonathan Guichard, Elayne Ruane, Ross Smith, Dan Bean, and Anthony Ventresque. 2019. Assessing the robustness of conversational agents using paraphrases. In *Proceedings of the International Conference On Artificial Intelligence Testing (AITest)*.
- [30] Futoshi Iwama and Takashi Fukuda. 2019. Automated testing of basic recognition capability for speech recognition systems. In *Proceedings of the Conference on Software Testing, Validation and Verification (ICST)*.
- [31] Stefano Lambiasi, Gemma Catolino, Fabio Palomba, and Filomena Ferrucci. 2024. Motivations, Challenges, Best Practices, and Benefits for Bots and Conversational Agents in Software Engineering: A Multivocal Literature Review. *Comput. Surveys* 57, 4 (2024), 1–37.
- [32] Amazon Lex. Accessed January 2026. *Amazon Lex Documentation*. <https://docs.aws.amazon.com/lex/>
- [33] Xiaomin Li, Chuanqi Tao, Jerry Gao, and Hongjing Guo. 2022. A Review of Quality Assurance Research of Dialogue Systems. In *Proceedings of the International Conference On Artificial Intelligence Testing (AITest)*.
- [34] Zixi Liu, Yang Feng, and Zhenyu Chen. 2021. DialTest: Automated testing for recurrent-neural-network-driven dialogue systems. In *Proceedings of the International Symposium on Software Testing and Analysis (ISSTA)*.
- [35] Google LLC. Accessed January 2026. *Dialogflow Release Notes*. <https://docs.cloud.google.com/dialogflow/docs/release-notes>
- [36] Jose María López-Morales, Pablo C Cañizares, Sara Pérez-Soler, Esther Guerra, and Juan de Lara. 2022. ASYMOB: A platform for measuring and clustering chatbots. In *Proceedings of the International Conference on Software Engineering, Companion (ICSE-C)*.
- [37] Elena Masserini, Diego Clerissi, Daniela Micucci, João Rodrigues Campos, and Leonardo Mariani. 2025. Towards the Assessment of Task-based Chatbots: From the TOFU-R Snapshot to the BRASATO Curated Dataset. In *Proceedings of the IEEE International Symposium on Software Reliability Engineering (ISSRE)*. Dataset available at <https://gitlab.com/securitychatbot/rasa-chatbot-dataset>
- [38] Rocco Gianni Rapisarda, Davide Ginelli, Diego Clerissi, and Leonardo Mariani. 2025. Test Case Generation for Dialogflow Task-Based Chatbots. In *Proceedings of the International Conference on Software Testing, Verification and Validation, Workshops (ICSTW)*.
- [39] Rasa. Accessed January 2026. *Rasa Documentation*. <https://rasa.com/docs/>
- [40] Elayne Ruane, Théo Faure, Ross Smith, Dan Bean, Julie Carson-Berndsen, and Anthony Ventresque. 2018. Botest: A framework to test the quality of conversational agents using divergent input examples. In *Proceedings of the International Conference on Intelligent User Interfaces, Companion (IUI-C)*.
- [41] Geovana Ramos Sousa Silva, Genaina Nunes Rodrigues, and Edna Dias Canedo. 2023. A modeling strategy for the verification of context-oriented chatbot conversational flows via model checking. *Journal of Universal Computer Science* 29, 7 (2023), 805.
- [42] Statista. Accessed January 2026. *The most spoken languages worldwide in 2025*. <https://www.statista.com/statistics/266808/the-most-spoken-languages-worldwide/>
- [43] Marisa Vasconcelos, Heloisa Candello, Claudio Pinhanes, and Thiago dos Santos. 2017. Bottester: Testing conversational systems with simulated users. In *Proceedings of the Brazilian Symposium on Human Factors in Computing Systems (IHC)*.