

Article

Not peer-reviewed version

AI Testing for Intelligent Chatbots – a Case Study

[Jerry Gao](#), [Radhika Agarwal](#)*, [Perna Garsole](#)

Posted Date: 21 February 2025

doi: 10.20944/preprints202502.1742.v1

Keywords: Chatbots; Smart AI Chat System Testing; 3D Intelligent Chat Test Modeling; Test Generation; Data Augmentation; AI test Result Validation



Preprints.org is a free multidisciplinary platform providing preprint service that is dedicated to making early versions of research outputs permanently available and citable. Preprints posted at Preprints.org appear in Web of Science, Crossref, Google Scholar, Scilit, Europe PMC.

Copyright: This open access article is published under a Creative Commons CC BY 4.0 license, which permit the free download, distribution, and reuse, provided that the author and preprint are cited in any reuse.

Article

AI Testing for Intelligent Chatbots—A Case Study

Jerry Gao ¹, Radhika Agarwal ^{2,*} and Prerna Garsole ³¹ Department of Computer Engineering, College of Engineering, San Jose State University, ALPSTouchStone, Inc., USA² ALPSTouchStone, Inc., California, USA³ Department of Computer Engineering, College of Engineering, San Jose State University, USA

* Correspondence: radhika6696@gmail.com

Abstract: Inspired by principles of the decision tree test method in software engineering, this paper provides a discussion on intelligent AI test modeling chat systems, including basic concepts, quality validation, test generation and augmentation, testing scopes, approaches, and needs. The novelty of the paper lies in an intelligent AI test modeling chatbot system that is built and implemented based on an innovative 3-dimensional AI test model for AI-powered functions in intelligent mobile apps to support model-based AI function testing, test data generation, and adequate test coverage result analysis.

Keywords: Chatbots; Smart AI Chat System Testing; 3D Intelligent Chat Test Modeling; Test Generation; Data Augmentation; AI test Result Validation

1. Introduction

An intelligent chat system refers to a computer-based intelligent chat system that has built-in AI solutions that support diverse system-and-user interactions with customers and answer questions, chat different subjects, and play as an intelligent agent (such as a finance agent, customer service agent, real estate agent, and so on). An intelligent chat system usually is developed with AI techniques, natural language processors, and machine models to accept, process, and understand diverse questions as inputs from users, generate appropriate responses to answer their questions, facilitate them to complete transactions, or walk users through a customer support process, and resolve their issues, typically represented as a virtual avatar.

Due to the advantages of intelligent chat systems (such as chatbots) over other customer support options, using chatbots is an effective method of customer engagement. It is becoming popular in business operations due to the advantages in the following areas: a) its easy connectivity with diverse social media channels and networks, such as websites, email, SMS, or messaging applications; b) cost reduction in call centers; c) easy collection of consumer data from support interactions; and d) smart interactions with customers.

According to a recent market analysis report [1], the chatbot market is expected to grow by USD 1.11 billion, progressing at a CAGR of almost 29% during the forecast period. Besides, organizations across various industry verticals are increasingly adopting AI to make more informed decisions. This provides enhanced customer service, opening several new opportunities for market vendors. In the academic field, comprehensive reviews on plenty of state-of-the-art research outcomes in dialogue systems [2] have been carried out, which is becoming a heated research topic.

This growing trend provides good business and research opportunities and brings technical challenges and needs in chatbot system testing and automation. As a special type of intelligent mobile app, testing smart chat systems must encounter similar challenges and issues [3]. For instance, smart mobile apps and modern intelligent chat systems have the following features:

- Development based on NLP and machine learning models based on big data.
- Rich-media inputs and outputs, text, audio, and images.
- Text-to-speech and speech-to-text functions.

- Text generation, synthesis, and analysis capabilities.
- Uncertainty and non-deterministic in response generation.
- Understanding selected languages and diverse questions and generating responses with diverse linguistic skills. These special features bring many issues and challenges in quality testing and evaluation of intelligent chat systems (like chatbots).

Certain issues [23] have been faced in testing intelligent chatbot systems as follows:

Issue 1 - Lack of quality validation criteria and quality assurance standards leads to difficulty in establishing well-defined, clear, and measurable quality testing requirements.

Issue 2 – Lack of well-defined systematic quality testing methods and solutions.

Issue 3 – Lack of automatic test tools with well-defined, adequate quality test coverage.

Issue 4 – Lack of automatic adequate quality test coverage analysis techniques and solutions.

The current chatbot testing methods do not consider some important features of intelligent chat systems, such as chatting patterns and steps, domain knowledge, memory, etc., which have been discussed in this work. The focus here is on various testing approaches used to validate intelligent chatbot systems and several non-functional quality testing services required for the system. Major causes of quality validation challenges include uncertainty, NLP/ML model diversity, continuous learning, platform diversity, big data diversity, and rich media inputs/outputs. To overcome these challenges and test for diversity, quality testing adequacy was carried out using some conventional testing methods. However, these methods were developed without considering AI-powered features and do not support rich media input. This led to the study of AI test modeling for intelligent chatbot systems.

The novelty of the paper lies in an intelligent chatbot-based 3-dimensional (input, context, and output) AI test modeling of the mobile app, Wysa, which is an emotional and mental coaching chatbot system. The main focus of the study is:

- Application of AI-based 3-dimensional test modeling and provides a reference classification test model for chatbot systems.
- Discussion of various test generation approaches for smart chatbot systems.
- AI to support test augmentation (positive and negative) for chatbots.
- Test result validation and adequacy using an AI-based approach for smart chatbot systems.
- AI test results and analysis, specifically for the mobile app, Wysa.

The paper discusses intelligent chat system testing, including basic concepts, testing scopes, approaches, and adequate validation criteria. In addition, the paper discusses 3D test modeling and provides a reference classification test model. The paper is structured as follows: Section 2 reviews the related research work on current smart chat systems. Section 3 discusses smart chat system testing, including basic concepts, the validation process, testing focuses, quality parameters, quality validation methods, and approaches. Section 4 discusses 3D intelligent test modeling and analysis with classified test modeling. Section 5 presents the test generation and data augmentation for intelligent chatbot systems. Section 6 reports test results and validation, with a statistical analysis of the results in Section 7. The concluding remarks of the work are presented in Section 8.

2. Literature Review

Many tools are available for chatbot development, but testing support for chatbots is very limited. Many existing chatbot development platforms, such as Dialogflow, Watson, or Lex, provide a web chat console that allows manual and informal testing of the chatbots. Only a few platforms, like Dialogflow, can provide debug facilities and check the quality of chatbots by detecting intents with similar training phrases. Some companies have developed testing tools for chatbots. For example, haptik.ai provides a testing tool that allows automatic interaction with chatbots through simple scripts and can be integrated with automation servers such as Jenkins. Botium can also be incorporated into testing flows using them.

For academic proposals, the authors introduced an approach for functional testing of a hotel booking chatbot and applied AI planning techniques to generate test cases traversing a conversation flow [9]. Divergent inputs (word order errors, incorrect verb tense, synonyms) were created from an initial utterance set [10], and divergent examples were generated by lexical substitutions [11] that retain the same meaning based on earlier studies. In recent years, there has been a surge of work on evaluating chatbots based on test models. Some researchers evaluated chatbots using natural language processing (NLP)-based approaches. Besides NLP-based test models, other recent test models can also be leveraged to test dialogue systems. The weakness of existing chatbot frameworks [12] was that different goals in diverse domain-oriented intelligent chat systems were not considered. To address this, the authors adapt the Goal-Question-Metric model [13], which is a top-down hierarchical model. The top level starts with a goal refined into several questions. Each question is a metric, objective, or subjective. The natural language conversations flow diagram (NLCFD)-based test execution [14] was then proposed as the set of specification traces, each of which is used to generate a set of test cases and then to repeatedly execute each test case until all associated possible paths have been covered.

Some authors have worked on the review of the quality assurance and test automation of the chatbot systems, including the works of [15–32]. Although these existing models have found various chatbot issues, a comprehensive test model is lacking in addressing the special test focuses and needs in domain knowledge, subjects, memory, diverse questions, and answers in the case of a mobile app chatbot system. This paper focuses on test modeling challenges and needs and provides a reference classification test model for the intelligent chatbot system, Wysa. The work provides several testing approaches that validate the intelligent chatbot system, followed by a non-functional quality testing service that helps improve its quality. The main aim of the study is to provide a 3D approach for AI test modeling, where the tree model includes input, context, and output classification. Table 1 represents a detailed survey based on AI test models of existing works and their differences in approach from the one studied here.

Table 1. Literature survey based on AI test models.

Ref	Objective	Automated test validation	Test Modelling	Test Generation	Augmentation
[4]	Testing conversational systems with simulated users (chatbot specialized in financial advice)	No	Simulated User Testing	Scenario-Based, and Behavior-Driven	Automated Testing Augmentation
[5]	Sequence-to-sequence models with long short-term memory (LSTM) for open-domain dialogue response generation	No	Dialogue generation model and personality model using OCEAN personality traits	Context-response pairs from two TV series (Friends and The Big Bang Theory)	Automatically introduces variations in the input
[6]	A metamorphic testing approach for chatbots and obtain sequences of interactions with a chatbot	No	Metamorphic testing	Metamorphic relations to guide the generation of test cases and initial set of inputs	Constructing grammars in Backus-Naur form (BNF), mutations, and functional correctness
[7]	Testing chatbots with Charm	No	Used Botium to test coherence, sturdiness, and precision testing	Convo Generation, Utterance Mutation	Fuzzing/Mutation, Iteration and Improvement
[8]	Turing test to AI chatbots to examine how chatbots behave in a suite of classic behavioral games using ChatGPT-4	No	Behavioral and personality testing	Classic behavioral games	Personality assessments using the Big-5 personality survey
Our purpose	Intelligent AI 3D test modeling, test generation, data augmentation, and test result validation for chat systems Wysa, a mental coach chatbot	Yes (Used AI-based test validation)	3-Dimensional AI test modeling	AI-based model, and NLP/ML model	AI- model based positive and negative augmentation

3. Understanding of Testing Intelligent Chatbot Systems

Testing intelligent chat systems refers to system quality validation activities using well-defined systematic methods and solutions to achieve the following three objectives:

Objective 1 - Validating the system chat intelligence and functions. It focuses on checking how well an intelligent chat system can interact with users to accept and correctly process incoming inputs in text/image/audio and generate the appreciated responses/answers. These interactive chats must be validated in a pre-defined domain-specific knowledge scope, including content subjects, topics, Q&A patterns, and interaction flows based on the predefined chat intelligence and functions at the system level.

Objective 2 - Measuring and assuring system nonfunction quality by evaluating its QoS parameters from different perspectives: a) language-oriented perception, diversity, and similarity; b) system-oriented parameters, such as security, reliability, scalability, availability, and performance, c) chatting related parameters, such as user satisfaction, response accuracy/relevance, content correctness, and consistency.

Objective 3 - Evaluating the system to see if it is trustworthy for users and customers based on user-oriented quality validation and evaluation.

3.1. Testing Approach

There are several testing approaches to validating intelligent chatbot systems. Table 2 compares AI testing, AI-based software testing, and conventional software testing methods.

Table 2. A comparison between AI testing, AI-based software testing, and the conventional software testing methods.

Items	AI Testing	AI-based Software Testing	Conventional Software Testing
Objectives	Validate and assure the quality of AI software and system by focusing on system AI functions and features	Leverage AI techniques and solutions to optimize a software testing process and its quality	Assure the system function quality for conventional software and its features
Primary AI-Testing Focuses	AI feature quality factors: accuracy, consistency, completeness, and performance	Optimize a test process in product quality increase, testing efficiency, and cost reduction	Automate test operations for a conventional software process
System Function Testing	AI system function testing: Object detection & classification, recommendation and prediction, language translation	System functions, behavior, user interfaces	System functions, behavior, user interfaces
Test Selection	AI test model-based test selection, classification and recommendation	Test selection, classification and recommendation using AI techniques	Rule-based and/or experience-based test selection
Test Data Generation	AI test model-based data discovery, collection, generation, and validation	AI-based data collection, classification, and generation	Model-based and/or pattern-based test generation
Bug Detection and Analysis	AI-model based bug detection, analysis, and report	Data-driven analysis for bug classification, detection, and prediction	Digital and systematic bug/problem management.

Different testing approaches are given below.

- **Conventional Testing Methods** - Using conventional software testing methods to validate any given smart chatbots and applications online or on mobile devices/emulators. These include scenario-based testing, boundary value testing, decision table testing, category partition testing, and equivalence partition testing.
- **Crowd-Sourced Testing** - Using crowd-sourced testers (freelanced testers) to perform user-oriented testing for given smart chatbots and systems. They usually use ad-hoc approaches to validate the given systems as a user.
- **Smart Chat Model-Based Testing** - Using model-based approaches to establish smart chat models to enable test case and data generation, and support test automation and coverage analysis.
- **AI-Based Testing** - Using AI techniques and machine learning models to develop AI-based solutions to optimize smart chatbot system quality test process and automation.
- **NLP/ML Model-Based Testing** - Using white-box or gray-box approaches to discover, derive, and generate test cases and data focusing on diverse ML models, related structures, and coverage.
- **Rule-based testing** - This methodology employs rule-based testing to generate tests and data for handling intelligent chat systems. While effective for traditional rule-based chat systems, it faces numerous challenges in addressing modern intelligent chat systems due to their unique characteristics and the complexities introduced by NLP-based machine learning models and big data-driven training.
- **System Testing** - In this strategy, quality of service (QoS) parameters at the system level will be chosen, assessed, and tested using clearly defined evaluation metrics. This aims to quantitatively validate both system-level functions and AI-powered features. Common system QoS factors encompass reliability, availability, performance, security, scalability, speed, correctness, efficiency, and user satisfaction. AI-specific QoS parameters typically involve accuracy, consistency, relevance, as well as loss and failure rates.
- **Learning-based testing** - In this method, the activities and tests conducted by human testers are monitored, and their test cases, data, and operational patterns are collected, analyzed, and learned

from to understand how they design tests and data for a specific smart chat system. Additionally, any bugs they uncover can be learned from and utilized as future test cases.

- **Metamorphic Testing** - Metamorphic testing (MT) is a property-based software testing technique, which can be an effective approach for addressing the test oracle problem and test case generation problem.

3.2. Testing Scope

The testing scope is shown in Figure 1 (a) for smart chat systems, including the following:

- **Domain Knowledge** - Validate a smart chatbot system to see how well the current chatbot system has been trained for selected domain knowledge and demonstrate its knowledge at different levels during its chats and communication with clients.
- **Chatflow Patterns** - Validate a smart chatbot system to see how well it is capable of carrying out chats in diverse chatting flow patterns.
- **Chat Memory** - Validate Chatbot's memory capability about clients' profiles, cases, chat history, and so on.
- **Q&A** - Validate Chatbot's Q&A intelligence to see how well an under-test smart chat system is capable of understanding the questions and responses from clients and providing the responses like a human agent.
- **Chat Subject** - Validate the Chatbot's intelligence in subject chatting to see how well an under-test chatbot system is capable of chatting with clients on selected subjects.
- **Language** - Validate a smart chatbot's language skills in communicating with clients in a selected natural language.
- **Text-to-speech** - Check how well the under-test chatbot system can convert text to speech correctly.
- **Speech-to-text** - Check how well the under-test chatbot system can convert speech audio to texts correctly.

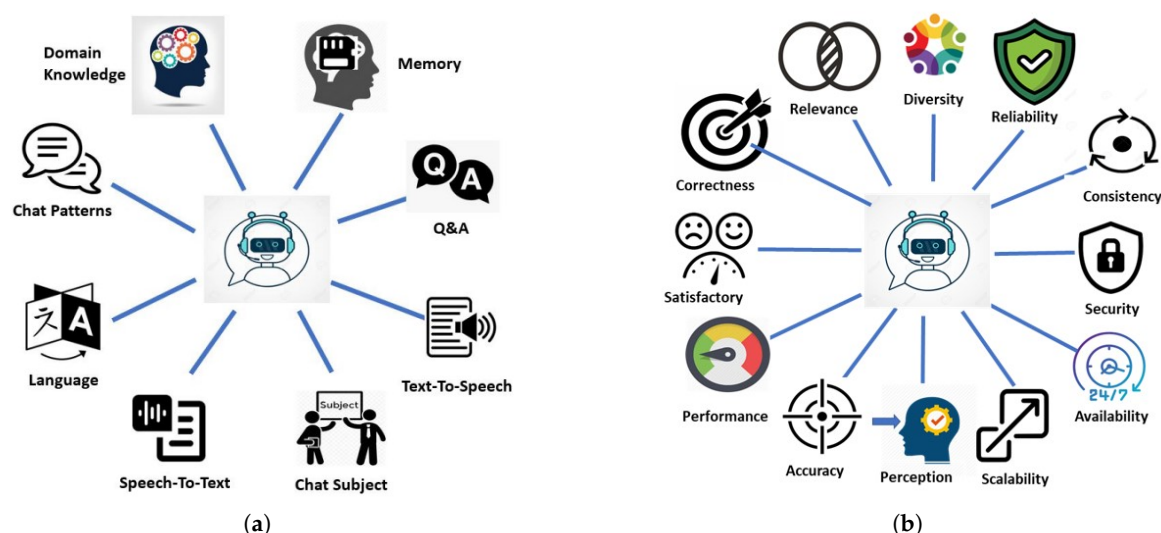


Figure 1. (a) Testing scope for intelligent chat functions. (b) Non-function testing scope for intelligent chatting.

The non-function quality testing service required for the system shown in Figure 1 (b) includes twelve quality validation focuses as follows:

- **Accuracy** - To make sure that the system can accurately process and understand NLP and rich media inputs, generate accurate chat responses in a variety of subject contents, languages, and linguistics using a range of tests with a variety of chat patterns and flows, assess system accuracy based on well-defined test models and accuracy metrics.

- **Consistency** - To ensure that the system can consistently process and understand a wide range of NLP, rich media inputs, and client attention, as well as generate consistent chat responses in a variety of subject contents, languages, and linguistics using a variety of tests with chat patterns and flows, evaluate system consistency based on well-defined test models and consistency metrics.
- **Relevance** - Use established testing models and profitability metrics to assess system relevance. This ensures the system's capability to comprehend and process a wide array of NLP and multimedia inputs, identify pertinent subjects and concerns, and produce appropriate chat responses across diverse domains, subjects, languages, and linguistic variations through varied testing methodologies.
- **Correctness** - Assess the accuracy of a chat system's processing and comprehension of NLP and/or rich media inputs, its ability to provide accurate replies related to domain subjects, contents, and language linguistics, using well-defined test models and metrics.
- **Availability** - Ensure system availability based on clearly defined parameters at various levels such as the underlying cloud infrastructure, the enabling platform environment, the targeted chat application SaaS, and user-oriented chat SaaS.
- **Security** - Assess the security of the system by utilizing specific security metrics to examine the chat system's security from various angles. This includes scrutinizing its cloud infrastructure, platform environment, client application SaaS, user authentication methods, and end-to-end chat session security.
- **Reliability** - Assess the reliability of the system by employing established reliability metrics at various tiers. This encompasses evaluating the reliability of the underlying cloud infrastructure, the deployed and hosted platform environment, and the chat application SaaS.
- **Scalability** - Evaluate system scalability based on well-defined scalability metrics in different perspectives, including deployed cloud-based infrastructure, hosted platform, intelligent chat application, large-scale chatting data volume, and user-oriented large-scale accesses.
- **User satisfactory** – Assess user satisfaction with the system by employing well-defined metrics from various angles. This includes analyzing user reviews, rankings, chat interactions, session success rates, and goal completion rates.
- **Linguistics diversity** – Assess the linguistic diversity of the intelligent chat system in its ability to support and process various linguistic inputs, including diverse vocabularies, idioms, phrases, and different types of client questions and responses. Additionally, evaluate the system's linguistic diversity in the responses it generates, considering domain content, subject matter, language syntax, semantics, and expression patterns.
- **Performance** - Assess the performance of the system using clearly defined metrics related to system and user response times, processing times for NLP-based and/or rich media inputs, and the time taken for generating chat responses.
- **Perception and understanding** – Assess the system's perception and comprehension of diverse inputs and rich media content from its clients using established test models and perception metrics.

4. AI Test Modeling for Intelligent Chatbot Systems

In software testing, the major objective of a test model is to provide a fundamental base for building systematic test methods and defining adequate test coverage assessment criteria. There is a strong demand for well-defined and practical test models for black-box validation of AI-powered functions in modern intelligent mobile apps and smart systems.

4.1. AI Test Modeling and Analysis

There are three ways to derive 3D AI test models for AI-powered functions, including manual derivation, tool-based interaction, and learning-based model derivation. The current version of the AI Test tool supports interactive test modeling for test engineers. Figure 2 shows the 3D AI function test model with $T(F_i)$ the AI function.

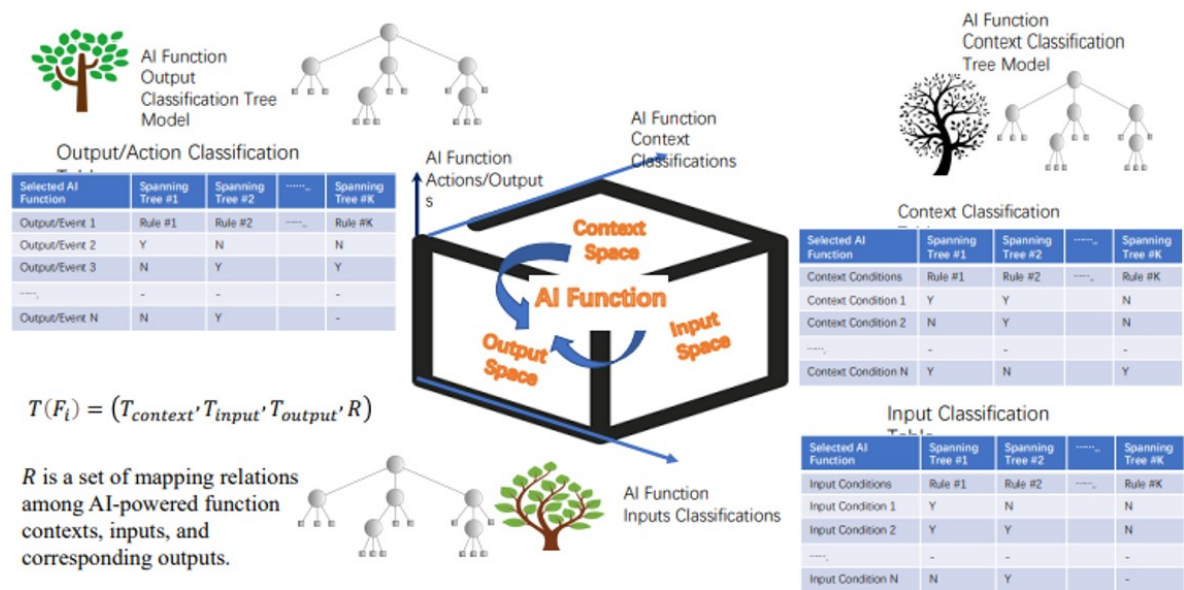
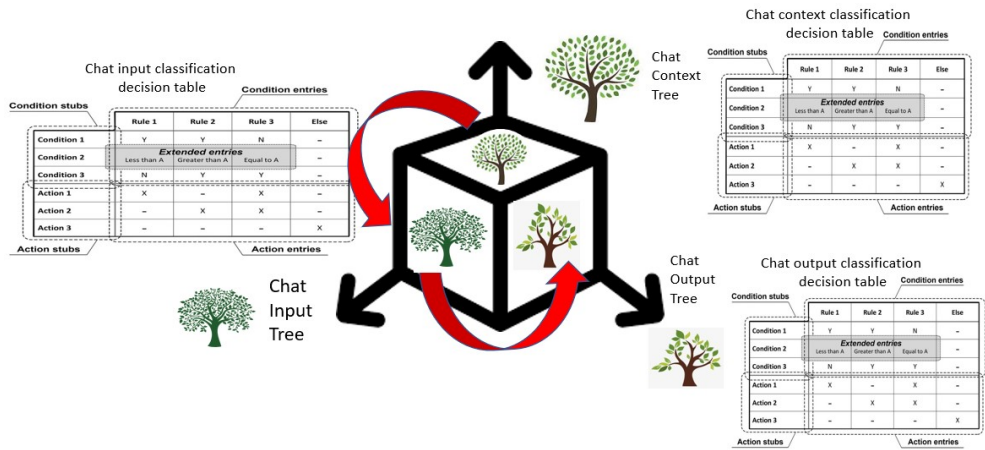


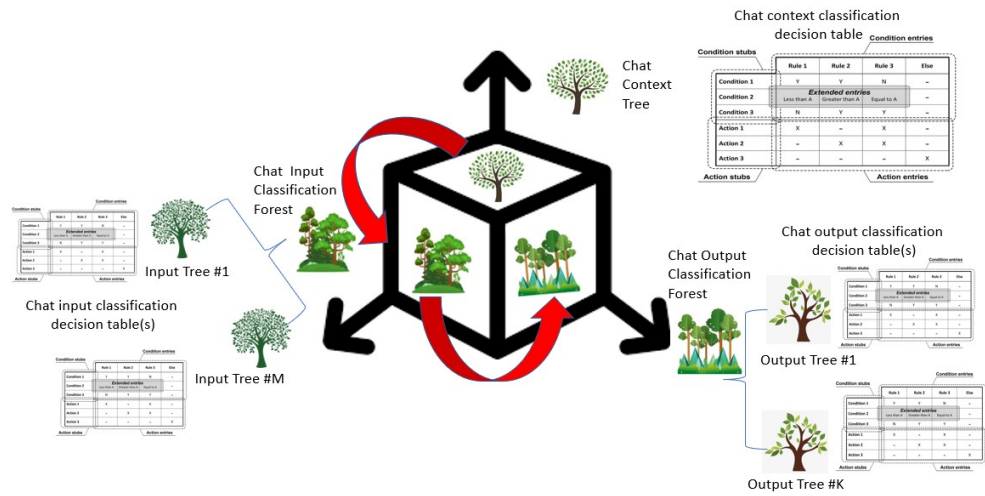
Figure 2. 3D AI function test model.

There are 3D intelligent chat test modeling processes discussed as follows:

- 3D Intelligent Chat Test Modeling Process I - In this process, for the selected intelligent chat function feature, set up a 3D classification tree model in 3 steps. 1) Define an intelligent chat input context classification tree model to represent diverse context classifications. 2) Define an intelligent chat input classification tree model to represent diverse chat input classifications. 3) Define an intelligent chat output classification tree model to represent diverse chat output classifications. Figure 3(a) shows the schematic diagram for a single-feature 3D classification tree model.
- 3D Intelligent Chat Test Modeling Process II - For the selected intelligent chat function features, set up a 3D classification forest tree model in 3 steps. 1) Define an intelligent chat input context classification tree model to represent diverse context classifications. 2) For each selected intelligent chat AI feature, define one intelligent chat input classification tree model to represent diverse chat input classifications. One input classification decision table is generated based on the defined input tree model. As a result, a chat input forest is created, and a set of input decision tables are generated. 3) For each selected intelligent chat AI feature, define one intelligent chat output classification tree model to represent diverse chat output classifications. One output classification decision table is generated based on the defined output tree model. As a result, a chat output forest is created, and a set of output decision tables is generated. Figure 3(b) shows the schematic diagram for the 3D classification forest tree model for selected features of intelligent chat functions.
- 3D Intelligent Chat Test Modeling Process III - For each selected intelligent chat function feature, set up a 3D classification tree model in 3 steps similar to that of process I. A corresponding 3D decision table will be generated. After a modeling process, a set of 3D tree models will be derived, and related 3D decision tables will be generated.



(a) A single-feature.



(b) For selected intelligent chat function features.

Figure 3. 3D classification tree model.

4.2. AI-Based Test Automation for Mobile Chatbot - WYSA

Wysa is an AI-powered coaching chatbot designed to provide mental health and emotional well-being support. It acts as a virtual companion and coach, offering a safe and confidential space for users to express their thoughts, feelings, and concerns. As an AI Coach, Wysa employs evidence-based techniques from cognitive-behavioral therapy (CBT), dialectical behavior therapy (DBT), mindfulness, and other therapeutic modalities to support users in managing stress, anxiety, depression, and other mental health challenges. Wysa offers a wide range of features and functionalities, including mood tracking, conversation analysis, personalized self-care exercises, goal setting, and coping strategies. It can help users identify and challenge negative thoughts, practice relaxation techniques, develop healthy habits, and set and track progress toward their well-being goals. Wysa AI feature scope is included in Table 3. Figure 4 shows the input, context, and output classification tree sample for the AI-based test automation for the mobile chatbot, Wysa.

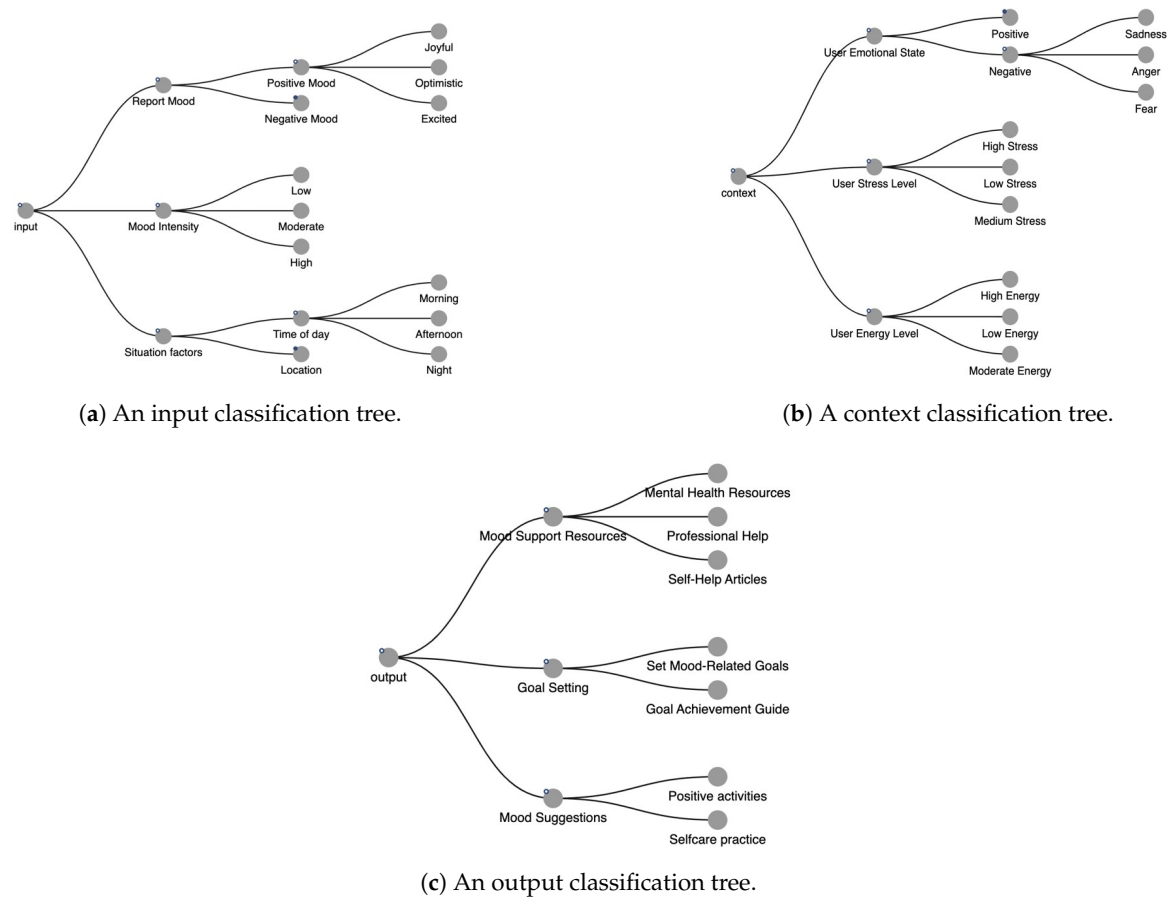


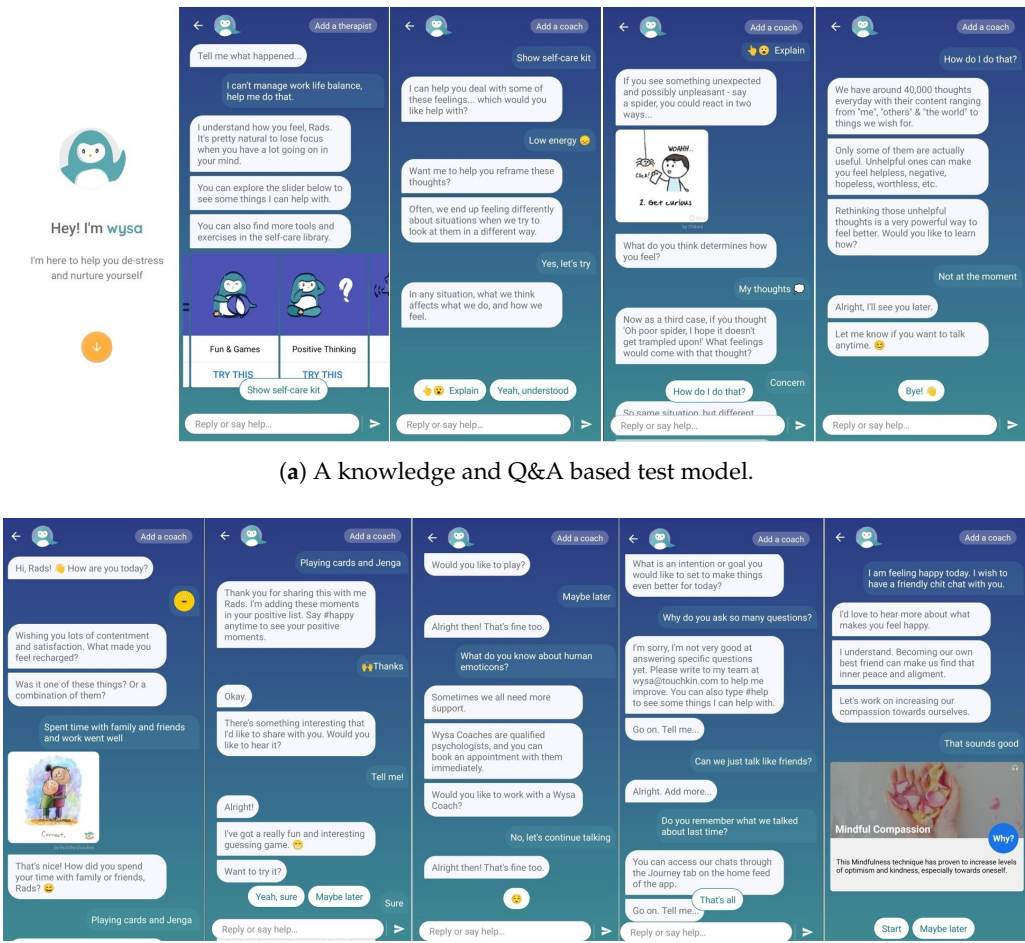
Figure 4. A sample of classification tree for AI-powered function in mobile app, Wysa.

Table 3. Wysa AI Feature Scope.

Concept	Description
Mood Tracking and Analysis	Tracking and analyzing an individual’s mood or emotional state over time. It involves recording and assessing mood patterns, fluctuations, and trends to gain insights into emotional well-being.
Self-care Analysis	Analyzing and evaluating an individual’s self-care practices and habits. It involves assessing activities related to physical, mental, and emotional well-being, such as exercise, sleep, relaxation techniques, and mindfulness practices.
Conversational Support Analysis	Analyzing the effectiveness and impact of conversational support provided by chatbots or AI systems. It involves evaluating the quality, empathy, and appropriateness of responses to users’ emotional or support-related queries or needs.
Goal Setting	Setting specific, measurable, attainable, relevant, and time-bound (SMART) goals to promote personal growth and well-being. It involves identifying areas of improvement, defining objectives, and establishing action plans to achieve desired outcomes.
Sentiment Analysis	Analyzing and determining the sentiment or emotional tone expressed in text or conversations. It involves classifying text as positive, negative, or neutral to understand the overall sentiment or attitude conveyed.
Well-being Resources and Personalised Intervention	Providing personalized resources, recommendations, or interventions to support individual well-being. It involves offering tailored suggestions, activities, or resources based on an individual’s needs, preferences, or identified areas of improvement.

4.3. Classified Test Modeling for Intelligent Chat System

Figure 5 shows the sample chat of the mobile chatbot system, Wysa, that works as a mental coach replying with different options and thoughts a person can work on for self-care. The chat shows the way Wysa tries to help a sad person uplift their mood by suggesting some links to mindful compassion. It also shows that Wysa remembers the person chatting and refers to accessing the journey tab for the previous chat. One can see the interactive pattern of Wysa. It tends to reply to short questions like 'tell me more', to W- questions like 'when, where, why, what', and knowledge questions based on analysis, application, or comprehension. It shows how Wysa tries to keep an interactive chat by suggesting a game.



(a) A knowledge and Q&A based test model.

(b) A memory and subject-oriented test model.

Figure 5. Chat sample for test modeling.

This subsection gives a classification test model for intelligent chat systems. It could support test design for the following types of intelligent chat testing.

- **Knowledge-based test modeling** - This is useful to establish test models focusing on selected domain-specific knowledge of a given intelligent chat system to see how well it understands the received domain-specific questions and provides proper domain-related responses relating to responses. For service-oriented domain intelligent chat systems, one could consider diverse knowledge of products, services, programs, and customers. Figure 6 (a) shows a knowledge-based test model, while (b) shows the specific example where American history is considered the knowledge domain and has topics and subtopics as shown in the figure.

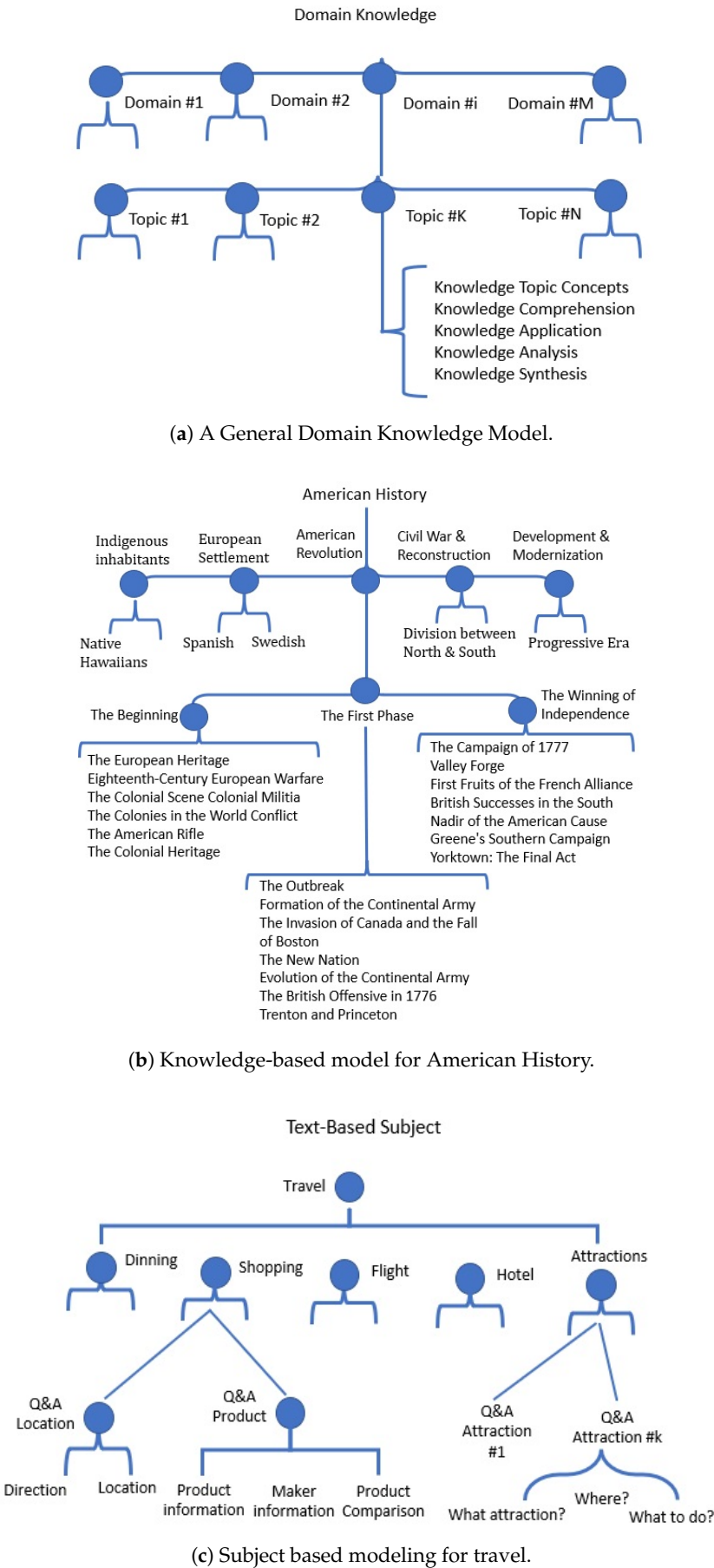


Figure 6. Knowledge and Subject-based Test Modeling.

- **Subject-oriented test modeling** - This is useful to establish test models focusing on a selected conversation subject for a given intelligent chat system to see how well it understands the questions and provides proper responses on diverse select subjects in intelligent chat systems. Typical conversational subjects include travel, driving directions, sports, and so on. Figure 6(c) shows an elaborate subject-based test modeling for travel and the certain queries that a person generally has about it.
- **Memory-oriented test modeling** - This is useful to establish test models for validating the memory capability of a given intelligent chat system to see how well it remembers users' profiles, questions, and related chats, as well as interactions. Figure 7 shows the text-based input tree for short and long-term memory.

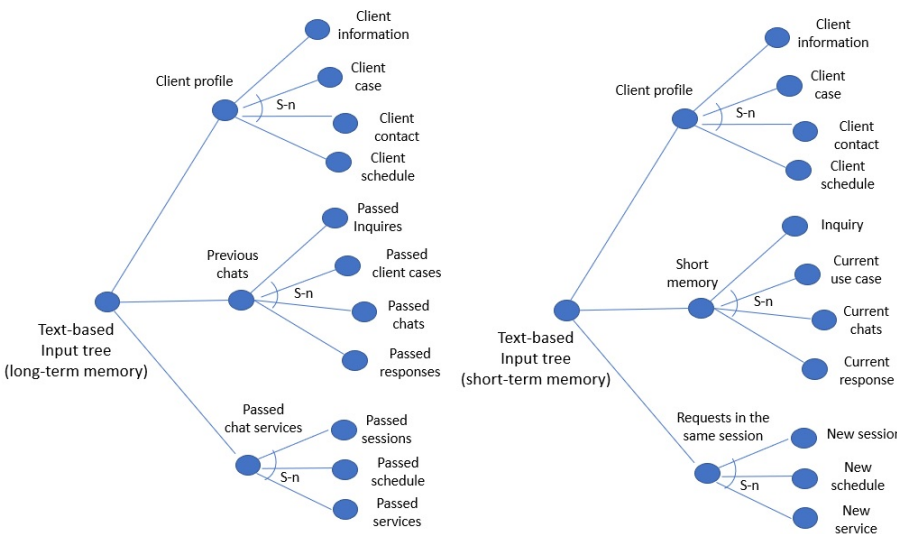


Figure 7. Memory-oriented Test Modeling.

- **Q&A pattern test modeling** - This is useful to establish test models for validating the diverse question and answer capability of a given intelligent chat system to see how well it handles diverse questions from clients and generates different types of responses. Figure 8 shows the text-based input tree for the question and answer pattern.

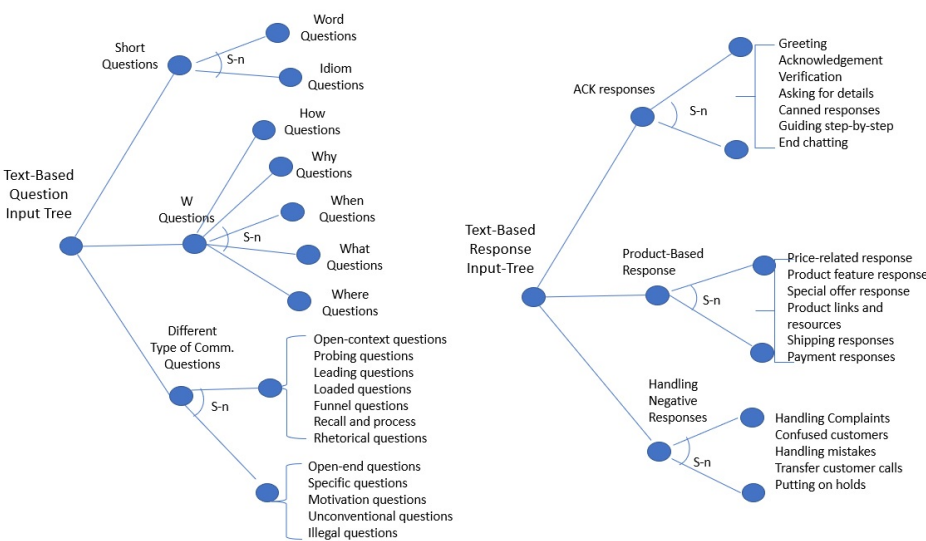


Figure 8. Q&A based Test Modeling.

- **Chat pattern test modeling** - This is useful to establish test models for validating diverse chat patterns for a given intelligent chat system to see how well it handles diverse chat patterns and interactive flows.
- **Linguistics test modeling** - This is useful to establish test models to validate language skills and linguistic diversity for a given intelligent chat system. Four aspects of test modeling for linguistics include sentences, diverse lexical items, different types of sentences, semantics, and syntax.

5. Test Generation and Data Augmentation for Intelligent Chatbot Systems

This section aims to discuss different ways to generate and augment the text and data for intelligent chatbot systems.

5.1. Test Generation

In this section, different test generation approaches for smart chatbot systems are discussed, namely, conventional, random, test model-based, AI-based, and NLP/ML models.

- **Conventional Test Generation** - It is simple and easy to use conventional software testing methods to generate tests for a selected smart chatbot system but there are certain limitations to this method, i.e., (1) difficult to perform test result validation using a systematic way, (2) hard to evaluate adequate test coverage, and (3) high costs to generate chat input tests manually.
- **Random Test Generation** - Using a random generator to select random chatbot tests as inputs from a given chatbot test DB to validate the under-test smart chatbot system. Random text generation can be used to generate unique and engaging text that can be used in various applications, from creative writing to marketing content.
- **Test Model Based Test Generation** - It is a model-based approach to enable automatic test generation for each targeted AI-powered intelligent chat function/feature. It supports model-based test coverage analysis and complexity assessment. It is important as AI-based test data augmentation solutions and result validation are needed. Figure 9 explains the semantic working of test model-based test generation.

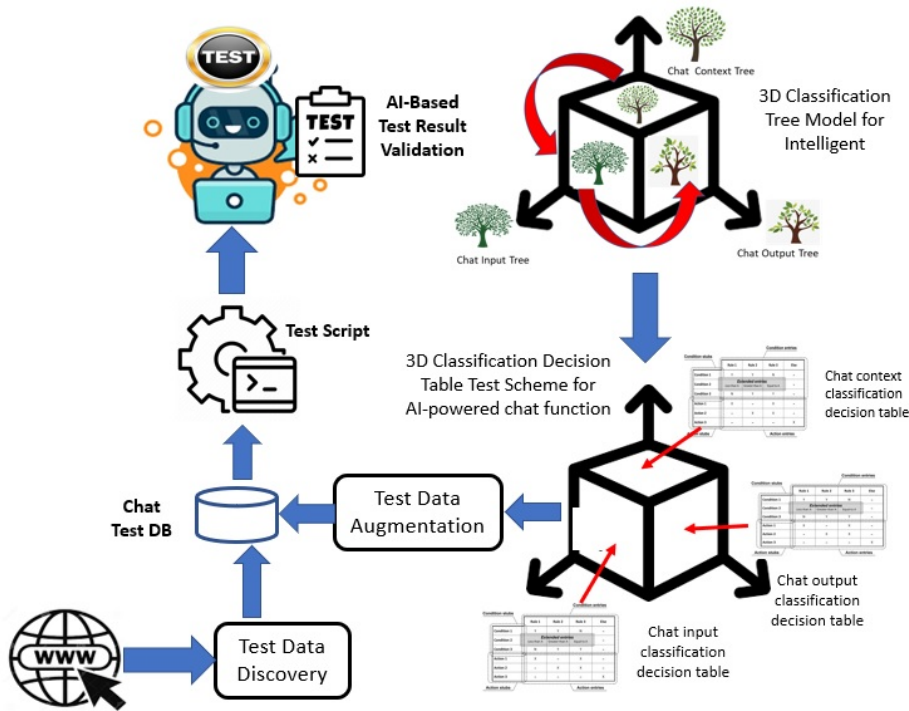


Figure 9. Test Model Based Test Generation.

- **AI-Based Test Generation** - Using AI techniques and machine learning models to generate augmented tests and test data for each selected test case. There are two types of data generated: 1) Synthetic data, which is generated artificially without using real-world images; and 2) Augmented data, which is derived from original images with some sort of minor geometric transformations (such as flipping, translation, rotation, or the addition of noise) to increase the diversity of the training set. This method also comes with its challenges, including: 1) Cost of quality assurance of the augmented datasets. 2) Research and Development to build synthetic data with advanced applications. 3) The inherent bias of original data persists in augmented data.
- **NLP/ML Model Test Generation** - Using white-box or gray-box approaches to discover, derive, and generate test cases and data based on each underlying NLP/ML model to achieve model-based structures and coverage.

5.2. Data Augmentation

Text augmentation is a technique used in natural language processing (NLP). It generates new examples of text by applying transformations to the original text data. The main purpose of text augmentation is to increase training dataset size and introduce variations in text. Generally, the common transformations seen here are synonym replacement, random insertion, random deletion, random swap, text masking, and character-level augmentation. It increases dataset diversity and enhances model performance and generalization in NLP tasks. They are used in text classification, sentiment analysis, named entity recognition, machine translation, etc. There are two types of augmentation techniques, namely, positive and negative text augmentation.

The positive augmentation comprises synonym and keyboard augmentation. (a) Synonym augmentation replaces words in a sentence with their synonyms. The purpose is to introduce lexical variation and expand the vocabulary used in the training data. By replacing words with their synonyms, the augmented data exposes the model to different word choices that convey similar meanings. This technique helps improve the model’s ability to handle diverse vocabulary and increases its flexibility in generating or understanding sentences with alternative word usage. (b) Keyboard augmentation simulates errors that can occur during manual typing on a keyboard. These errors can include accidental character swaps, deletions, or insertions, often caused by typographical mistakes or the proximity of keys on a keyboard. By introducing such errors into the text data, keyboard augmentation helps the model learn to handle and correct these types of errors. It improves the model’s ability to recognize and interpret text data with typographical variations, enhancing its robustness in real-world scenarios.

The negative augmentation comprises random word swap augmentation, random word delete augmentation, Optical Character Recognition (OCR) augmentation, random char swap augmentation, and random char insert augmentation. Table 4 shows the example of different negative augmentation.

Table 4. Negative Data Augmentation

Text Input	Augmentation	Text Output
I am Sad	random char insert	I am aSad
I am Sad	random char swap	I am Sda
I am Sad	random char delete	I am ad
I am Sad	random word swap	Sad am I
I am Sad	random word delete	am Sad
I am Sad	ocr augmentation	1 am Sad

6. AI-Based Test Result Validation and Adequacy Approaches for Smart Chatbot Systems

1. **Conventional Testing Oracle** - Test Oracle is a mechanism that can be used to test the accuracy of a program’s output for test cases. Conceptually, consider testing a process in which test cases are given for testing and the program under test. The output of the two is then compared to

determine whether the program behaves correctly for test cases. Ideally, an automated oracle is required, which always gives the correct answer. However, often oracles are human beings, who mostly calculate manually the output of the program. Consequently, when there is a discrepancy, between the program and the result, we must verify the result produced by the oracle before declaring that there is a defect in the result.

2. **Text-Based Result Similarity Checking** - For text-based chat outputs, AI-based approaches can be used to perform text similarity analysis to determine if the intelligent chat test results are acceptable. Figure 10 (a) and (b) shows the language-based similarity evaluation. Lexical similarity is a measure of two texts' similarity based on the intersection of word sets from the same or different languages. Lexical similarity scores range from 0 to 1, indicating no common terms between the two texts, to 1, indicating total overlap between the vocabulary. Figure 10 (c) shows the second approach, keyword-based weighted text similarity evaluation with the similarity formula.

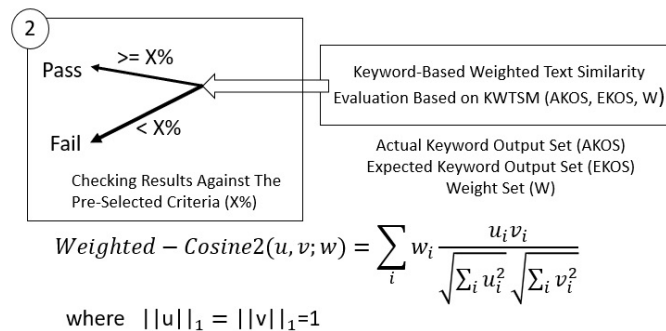
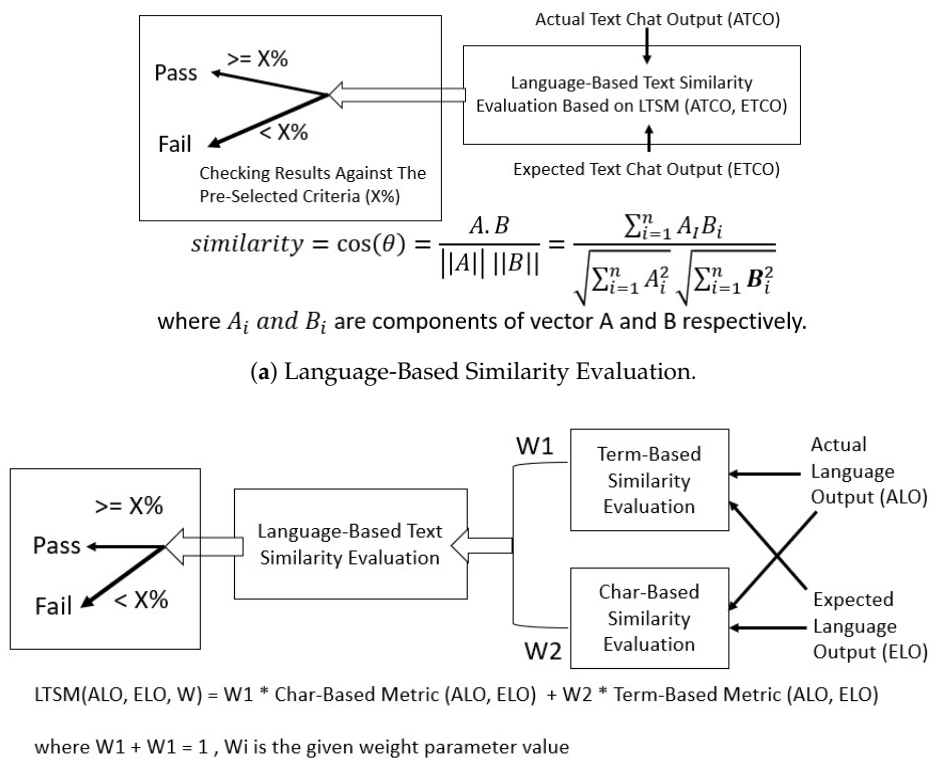


Figure 10. Similarity evaluation.

3. **Image-Based Similarity Checking** - For image-based outputs, AI-based approaches can be used to perform image similarity analysis to determine if the intelligent chat test results are acceptable. Diverse machine learning algorithms and deep learning models can be used to compute the similarity at the different levels and perspectives between an expected output image and a real output image from the under-test chatbot system, including objects/types, features, positions, scales, and colors.
4. **Audio-Based Result Similarity Checking** - For audio-based outputs, AI-based approaches can be used to perform radio similarity analysis to determine if the intelligent chat test results are acceptable. Diverse machine learning algorithms and deep learning models can be used to compute the similarity (at different levels and perspectives) between an expected output audio and real output audio from the under-test chatbot system, including audio sources/types, audio features, timing, frequencies, noises, and so on.

6.1. Model-Based Test Coverage for Smart Chatbot system

One can establish a 3D AI test classification decision table for each targeted under-test computer vision intelligence based on the established 3D AI tree model. Using this test classification decision table as a test scheme, a set of 3D AI test classification test case sets (known as 3DT-Set) could be generated. The four test coverage criteria can be defined below:

1. 3-dimensional AI Test Classification Decision Table Test Coverage - To achieve this coverage, the test set (3DT-Set) must include one test case for any 3D element - T (CT-x, IT-y, OT-z) in the 3D AI test classification decision table.
2. Context classification decision table test coverage - To achieve this coverage, the test set(3DT-Set) must include one test case for any rule in a context classification decision table.
3. Input classification decision table test coverage - To achieve this coverage, the test set (3DT-Set) must include at least one test case for any rule in an input classification decision table.
4. Output classification decision table test coverage - To achieve this coverage, the test set (3DT-Set) must include at least a tree case for any rule in an output classification decision table.

Here, the intent is only to provide various coverage criteria. This paper focuses on test modeling and data generation, so a detailed explanation and an example of coverage criteria will be discussed in future studies.

7. AI Test Result and Analysis

Wysa uses natural language processing (NLP) to understand and respond to messages in real time. It provides access to a library of mental health resources, including articles, videos, and guided meditations. Wysa provides insights into the user's mental health patterns while tracking the user's mood over time. It offers exercises and techniques supported by the facts to assist you in managing various mental health issues and provides a variety of chat modes, such as introspective, coaching, and conversational, to meet the user's needs and preferences. A built-in gratitude journal and mood tracker are included to assist users in developing a positive mindset and monitoring their growth.

This section presents four different testing criteria, including general responses, memory-based responses, emotive reflexes, and Q&A-based interaction performed on Wysa, and its performance during manual and automated testing.

7.1. General Responses

By creating domain-driven tests and test models, this test aims to determine how well a chat system can handle chat questions and answers related to a specific field based on domain-driven validation. This would assist in identifying and evaluating the functionality of an application. Figure 11 shows the 3D diagram for the general responses.



Figure 11. 3D diagram for the General Responses.

The app is strengthened, and its intended activities are emphasized. Figure 12 shows the test case results of manual and automation testing for the general chat responses with their respective pie-chart representations and bar-chart for multiple scenarios.

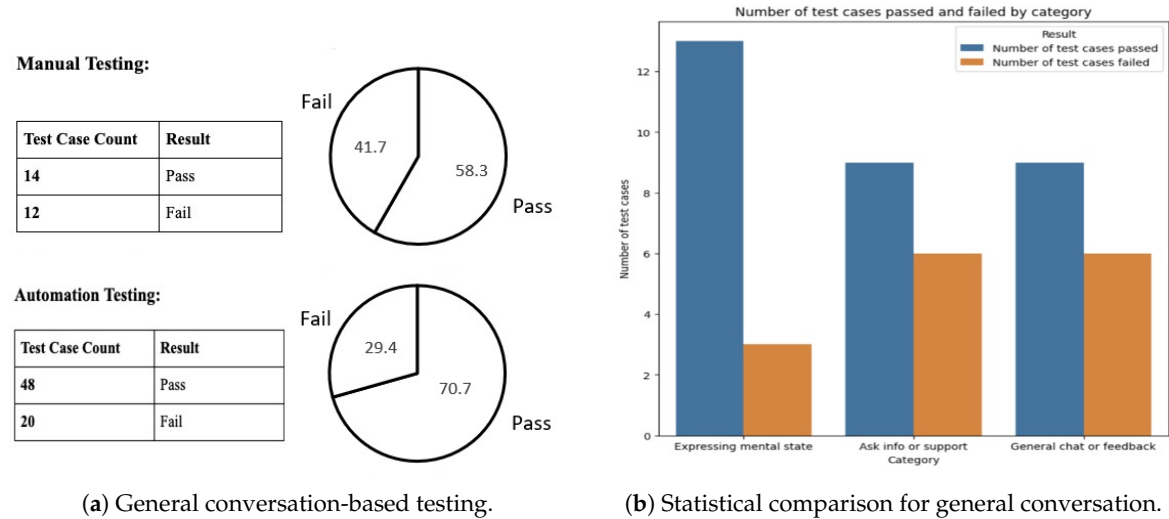


Figure 12. General Response-based Testing and Analysis.

7.2. Memory-Based Responses

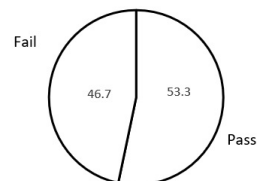
The chatbot must remember the context of the conversation and user preferences. Previously discussed chats were used to answer a current question. It was tested to predict the outcome with information and questions. Figure 13 shows the 3D diagram for the general responses. A variety of scenarios were considered to assess both the chatbot’s past and present memory. For a chatbot like WYSA, memory testing is crucial, as it must remember user preferences and conversations to properly respond to what the user has been going through. Figure 14 shows the test case results of manual and automation testing for the memory-based responses.



Figure 13. 3D diagram for the General Responses.



Test Case Count	Result
16	Pass
14	Fail

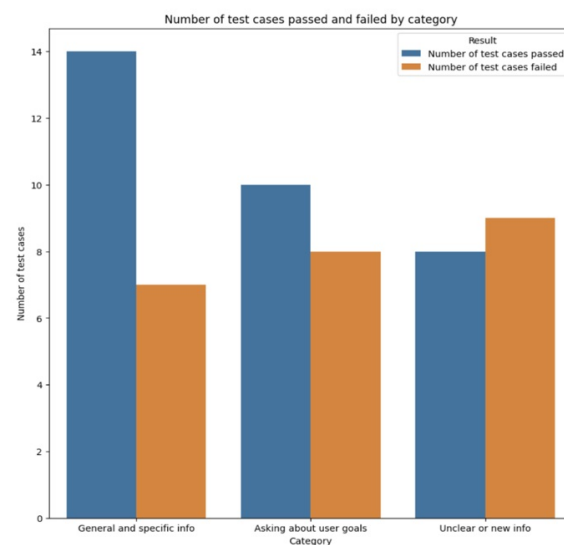


Automation Testing:

Test Case Count	Result
24	Pass
17	Fail



(a) Memory-based testing.



(b) Statistical comparison for Memory testing.

Figure 14. Memory-based Testing and analysis.

7.3. Emotive Reflexes

Wysa application uses AI technology to test and analyze these emotive reflexes, enabling users to better comprehend and manage their emotions. One of the key features of Wysa is its Emotional Reflexes Test area, which is designed to help users better understand their emotional reactions to different situations. This mental health chatbot makes use of a combination of cognitive-behavioral therapy (CBT), meditation, and mindfulness techniques to help users manage stress, anxiety, and depression. Figure 15 shows the test case results of manual and automation testing for the emotional responses of the chatbot with their respective pie-chart representations and bar-charts for multiple scenarios.

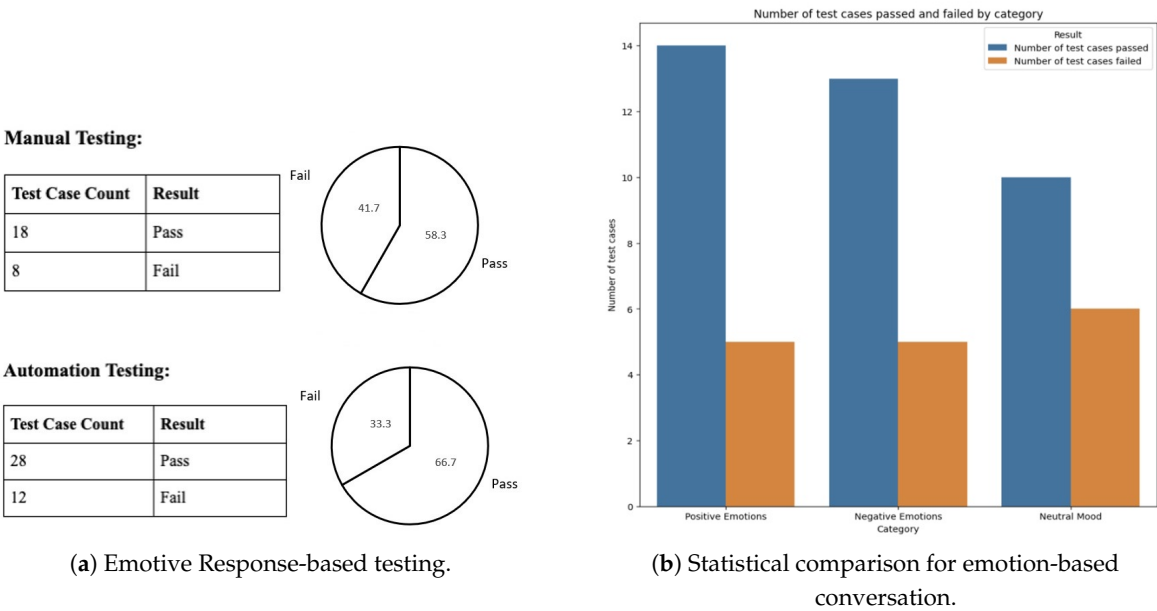


Figure 15. Emotive Reflex-based Testing and Analysis.

A sample in Figure 16 shows the bug report, where the AI chatbot could not read the emoji and respond with an appropriate message. So chatbots tend to be in the learning process with larger inputs.

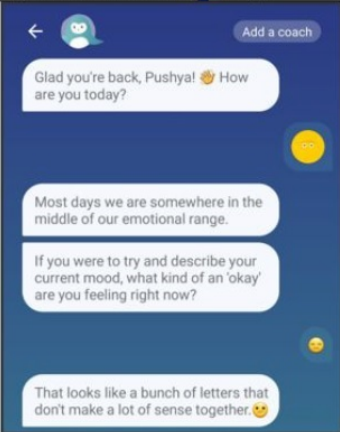
Test Description	Testing on Emoji inputs
Steps to Reproduce	Step-1: Open Wysa Application Step-2: Open Chatbot from the Self-Care Section Step-3: Respond to the chatbot about your feeling with sad by dragging the smile on the screen. Step-4: Send a message to the chatbot as a emoji.
Attached Proof	
Expected Result	The chatbot reads the emoji and responds with appropriate message.
Actual Result	It appears to be a collection of letters that don't make any sense when put together.
Test Result	Failed

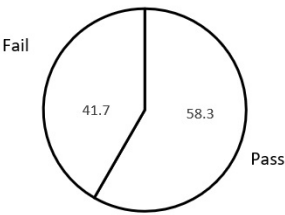
Figure 16. Sample of a bug report for emotive reflexes.

7.4. Q&A Interactions

In addition to responding to domain-relevant requests, the chatbot should also be able to answer domain-specific queries. It is possible to measure the intelligence of a chatbot by its ability to respond to complex, comprehensive questions using its memory and problem-solving skills. In addition to comprehensive questions, there are also short questions, loaded questions, and analytical questions. Figure 17 shows the test case results of manual and automation testing concerning the Q&A-based chat responses with their respective pie-chart representations and bar-chart for multiple scenarios.

Manual Testing:

Test Case Count	Result
14	Pass
10	Fail

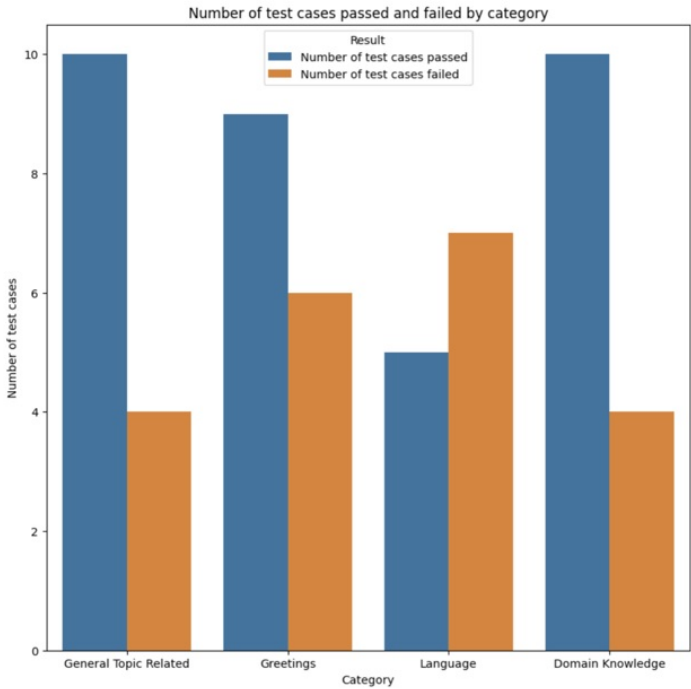


Automation Testing:

Test Case Count	Result
27	Pass
16	Fail



(a) Q&A-based testing.



(b) Statistical comparison for Q&A based conversation.

Figure 17. Q&A-based Testing and analysis.

8. Conclusion and Future Scope

With the fast acceptance of intelligent chatbots and applications in customer support, more and more intelligent mobile chatbots are being deployed in diverse customer services and applications. This requires adequate quality testing, modeling, and test automation platforms and solutions. This paper provides a case study on mental health and emotional well-being supporting intelligent chatbot systems, Wysa, from different perspectives, including 3-dimensional modeling (input, context, and output), test generation, data augmentation, and test validation. It helps track and analyze mood, provides conversational support, and helps in sentiment analysis. As a part of the analysis, the work includes general responses, memory-based responses, and passionate responses for Wysa (intelligent chatbot system). Also, it concludes that such systems are under testing and still have bugs that are being resolved continuously.

In the future, work can be done to build the tools to support chatbot automation and test data augmentation. These different types of automation tools can simulate crime-based chatbots and call-based chat systems to support test modeling. One can also refer to this work for intelligent computer vision systems where AI-test modeling can be used to study image and document-based intelligence.

Author Contributions: Jerry Gao: Conceptualization, formal analysis, resources, supervision, review, and administration; Radhika Agarwal: original draft preparation, validation, case study, writing-review, and editing; Prerna Garsole: data curation, methodology, software, validation, formal analysis. All authors have read and agreed to publish the paper in this reputed journal.

Funding: This research received no external funding.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Businesswire, "Global Chatbot Market Value to Increase by \$1.11 Billion during 2020-2024 | Business Continuity Plan and Forecast for the New Normal | Technavio". Available: <https://www.businesswire.com/news/home/20201207005691/en/Global-Chatbot-Market-Value-to-Increase-by-1.11-Billion-during-2020-2024-Business-Continuity-Plan-and-Forecast-for-the-New-Normal-Technavio>.
2. J. Ni, T. Young, V. Pandealea, F. Xue, V. Adiga, and E. Cambria, "Recent Advances in Deep Learning-based Dialogue Systems," *ArXiv210504387 Cs*, 2021. <http://arxiv.org/abs/2105.04387>
3. C. Tao, J. Gao, and T. Wang, "Testing and Quality Validation for AI Software—Perspectives, Issues, and Practices," *IEEE Access*, vol. 7, pp. 120164–120175, 2019. <https://doi.org/10.1109/ACCESS.2019.2937107>.
4. 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, pp. 1–4, 2017. <https://doi.org/10.1145/3160504.3160584>.
5. Y. Xing and R. Fernández, "Automatic Evaluation of Neural Personality-based Chatbots," in *Proceedings of the 11th International Conference on Natural Language Generation*, Tilburg University, The Netherlands, pp. 189–194, 2018. <https://doi.org/10.18653/v1/W18-6524>.
6. J. Bozic, and F. Wotawa, "Testing Chatbots Using Metamorphic Relations," in Gaston, C., Kosmatov, N., Le Gall, P. (eds) *Testing Software and Systems*, ICTSS 2019. Lecture Notes in Computer Science, vol. 11812. Springer, Cham, 2019. https://doi.org/10.1007/978-3-030-31280-0_3
7. S. Bravo-Santos, E. Guerra, and J. de Lara, "Testing Chatbots with Charm," in M. Shepperd, F. Brito e Abreu, A. Rodrigues da Silva, and R. Pérez-Castillo, (eds) *Quality of Information and Communications Technology*, QUATIC 2020. Communications in Computer and Information Science, vol 1266. Springer, Cham. doi: 10.1007/978-3-030-58793-2_34
8. Q. Mei, Y. Xie, W. Yuan, and M. O. Jackson, "A Turing test of whether AI chatbots are behaviorally similar to humans," *Economic Sciences*, vol. 121, no. 9, e2313925121, 2024, doi: 10.1073/pnas.2313925121
9. J. Bozic, O. A. Tazl, and F. Wotawa, "Chatbot Testing Using AI Planning," in *Proceedings of the 2019 IEEE International Conference On Artificial Intelligence Testing (AITest)*, Newark, CA, USA, pp. 37–44, 2019. <https://doi.org/10.1109/AITest.2019.00-10>.
10. E. Ruane, T. Faure, R. Smith, D. Bean, J. Carson-Berndsen, and A. Ventresque, "BoTest: a Framework to Test the Quality of Conversational Agents Using Divergent Input Examples," in *Proceedings of the 23rd International Conference on Intelligent User Interfaces Companion*, Tokyo Japan, pp. 1–2, 2018. <https://doi.org/10.1145/3180308.3180373>.
11. J. Guichard, E. Ruane, R. Smith, D. Bean, and A. Ventresque, "Assessing the Robustness of Conversational Agents using Paraphrases," *Proceedings of the 2019 IEEE International Conference On Artificial Intelligence Testing (AITest)*, Newark, CA, USA, pp. 55–62, 2019. <https://doi.org/10.1109/AITest.2019.000-7>.
12. M. Kaleem, O. Alobadi, J. O'Shea, and K. Crockett, "Framework for the formulation of metrics for conversational agent evaluation," in *RE-WOCHAT: Workshop on Collecting and Generating Resources for Chatbots and Conversational Agents-Development and Evaluation Workshop Programme*, pp. 20–23, 2016.
13. M. Nick and C. Tautz, "Practical evaluation of an organizational memory using the goal-question-metric technique," in *Proceedings of Biannual German Conference on Knowledge-Based Systems*, pp. 138–147, 1999.
14. M. Padmanabhan, "Test Path Identification for Virtual Assistants Based on a Chatbot Flow Specifications," In: K. N. Das, J. C. Bansal, K. Deep, and A. K. Nagar, P. Pathipooranam, and R. C. Naidu, *Soft Computing*

- for Problem Solving. *Advances in Intelligent Systems and Computing*, Springer, Singapore, vol. 1057, 2020, doi: 10.1007/978-981-15-0184-5_78.
15. F. Aslam, "The impact of artificial intelligence on chatbot technology: A study on the current advancements and leading innovations," *European Journal of Technology*, vol. 7, no. 3, pp. 62–72, 2023.
 16. S. Ayanouz, B. A. Abdelhakim, and M. Benhmed, "A smart chatbot architecture based NLP and machine learning for health care assistance," in *Proceedings of the 3rd International Conference on Networking, Information Systems, & Security*, pp. 1–6, 2020.
 17. S. Bialkova, "Chatbot Efficiency—Model Testing," in *The Rise of AI User Applications*, Springer, Cham, 2024, doi: 10.1007/978-3-031-56471-0_5
 18. G. Bilquise, S. Ibrahim, and K. Shaalan, "Emotionally intelligent chatbots: A systematic literature review," *Human Behavior and Emerging Technologies*, vol. 2022, 2022.
 19. G. Caldarini, S. Jaf, and K. McGarry, "A Literature Survey of Recent Advances in Chatbots," *Information*, vol. 13, no. 1, pp. 41, 2022, doi: 10.3390/info13010041
 20. J. Gao, C. Tao, D. Jie, and S. Lu, "Invited Paper: What is AI Software Testing? and Why," 2019 *IEEE International Conference on Service-Oriented System Engineering (SOSE)*, San Francisco, CA, USA, pp. 27–2709, 2019, doi: 10.1109/SOSE.2019.00015.
 21. J. Gao, P. H. Patil, S. Lu, D. Cao and C. Tao, "Model-Based Test Modeling and Automation Tool for Intelligent Mobile Apps," 2021 *IEEE International Conference on Service-Oriented System Engineering (SOSE)*, Oxford, United Kingdom, pp. 1–10, 2021, doi: 10.1109/SOSE52839.2021.00028.
 22. J. Gao, S. Li, C. Tao, Y. He, A. P. Anumalasetty, E. W. Joseph, and H. Nayani, "An approach to GUI test scenario generation using machine learning," In 2022 *IEEE International Conference on artificial intelligence testing (AITest)*, pp. 79–86, 2022.
 23. J. Gao, P. Garsole, R. Agarwal and S. Liu, "AI Test Modeling and Analysis for Intelligent Chatbot Mobile App - A Case Study on Wysa," in 2024 *IEEE International Conference on Artificial Intelligence Testing (AITest)*, Shanghai, China, 2024, pp. 132–141, doi: 10.1109/AITest62860.2024.00024.
 24. M. H. Kurniawan, H. Handiyani, T. Nuraini, R. T. S. Hariyati, and S. Sutrisno, "A systematic review of artificial intelligence-powered (AI-powered) chatbot intervention for managing chronic illness," *Annals of Medicine*, vol. 56, no. 1, 2302980, 2024.
 25. X. Li, C. Tao, J. Gao and H. Guo, "A Review of Quality Assurance Research of Dialogue Systems," in 2022 *IEEE International Conference On Artificial Intelligence Testing (AITest)*, Newark, CA, USA, pp. 87–94, 2022, doi: 10.1109/AITest55621.2022.00021.
 26. C.-C. Lin, A.Y.Q. Huang, and S.J.H. Yang, "A Review of AI-Driven Conversational Chatbots Implementation Methodologies and Challenges (1999–2022)," *Sustainability*, vol. 15, no. 2, 2023, doi: 10.3390/su15054012.
 27. E. W. Ngai, M. C. Lee, M. Luo, P. S. Chan, and T. Liang, "An intelligent knowledge-based chatbot for customer service," *Electronic Commerce Research and Applications*, vol. 50, 101098, 2021.
 28. A. Park, S. B. Lee, and J. Song, "Application of AI based Chatbot Technology in the Industry," *Journal of the Korea Society of Computer and Information*, vol. 25, no. 7, pp. 17–25, 2020.
 29. D. M. Park, S. S. Jeong, and Y. S. Seo, "Systematic review on chatbot techniques and applications," *Journal of Information Processing Systems*, vol. 18, no. 1, pp. 26–47, 2022.
 30. C. Tao, J. Gao and T. Wang, "Testing and Quality Validation for AI Software—Perspectives, Issues, and Practices," in *IEEE Access*, vol. 7, pp. 120164–120175, 2019, doi: 10.1109/ACCESS.2019.2937107.
 31. C. T. P. Tran, M. G. Valmiki, G. Xu, and J. Z. Gao, "An intelligent mobile application testing experience report," in *Journal of Physics: Conference Series*, vol. 1828, no. 1, pp. 012080, 2021. IOP Publishing.
 32. L. Xu, L. Sanders, K. Li, and J. C. Chow, "Chatbot for health care and oncology applications using artificial intelligence and machine learning: systematic review," *Journal of Medical Internet Research (JMIR) cancer*, vol. 7, no. 4, e27850, 2021.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.