Research Article

Jathushan Vivekananthan*, Mian Usman Sattar, and Maximilian Lackner*

Artificial intelligence for enhanced quality assurance through advanced strategies and implementation in the software industry

https://doi.org/10.1515/jisys-2024-0377
received August 15, 2024; accepted December 9, 2024

Abstract: In this study, an artificial intelligence (AI) assistant is developed specifically for quality assurance (QA) tasks in response to the software industry's growing demand for better QA solutions. Traditional QA methods are labor-intensive and prone to human error, even when they function properly. Utilizing the most recent advancements in natural language processing and AI, this AI assistant maximizes output and reliability by automating and optimizing QA processes. Using Rasa technology, the AI assistant aims to revolutionize QA by offering testers comprehensive support, including question-answering, contextual recommendations, and expanding users' knowledge bases. It can reply to queries using both words and images. They are minimizing the amount of time spent searching for answers and raising user engagement. Notwithstanding AI's potential to improve quality control, challenges remain, particularly in the area of human-like language production and comprehension. In today's industry, professional QA tester training is a big worry. Due to the conventional training method's substantial dependence on senior testers to educate and assist trainees, only two or three trainees may typically obtain appropriate instruction at a time from a single senior QA tester. This limitation affects the scalability of QA training programmers. With the new approach, QA training can be made much more productive and scalable since a single senior QA tester can now educate over 20 new hires at once.

Keywords: quality assurance, artificial intelligence, natural language processing, Rasa core, AI assistant, chatbot

1 Introduction

Software quality assurance (QA) is an integral part of software industrial operations; it checks that products and services meet pre-determined standards. With changes in industries, more effective and reliable QA solutions are required. Even though traditional QA techniques work, they are often time- and manpower-intensive. The evolution of natural language processing (NLP) and artificial intelligence (AI) has created new opportunities for QA mechanisms. This article presents the design of an AI assistant for the QA field. The recommended AI assistant utilizes state-of-the-art technologies such as NLP and Rasa technology to offer full assistance to QA testers [1]. Rasa platform is the leading open generative conversational AI framework for building and delivering next-level AI assistants. This includes answering their questions, giving them advice that caters to exactly where they work, and the overall expansion of knowledge. AI assistants vastly

*** Corresponding author: Jathushan Vivekananthan,** Department of Computing, College of Science and Engineering, University of Derby, DE21 1GB Derby, United Kingdom, e-mail: j.vivekananthan1@unimail.derby.ac.uk
*** Corresponding author: Maximilian Lackner,** Department of Industrial Engineering, University of Applied Sciences Technikum Wien, Hoechstaedtplatz 6, 1200 Vienna, Austria, e-mail: maximilian.lackner@technikum-wien.at
Mian Usman Sattar: Department of Computing, College of Science and Engineering, University of Derby, DE21 1GB Derby, United Kingdom, e-mail: u.sattar@derby.ac.uk

underestimate the complexity of having to generate and understand language in such a human-like way. While single-word disambiguation is one part of the puzzle, it has been shown that interpreting cohesive phrases when words are used in a sentence also makes computer comprehension poor [2]. Chatbots, conversely, are virtual talking devices that have conversations just like a real human. Chatbots are also referred to as robotic virtual agents [3]. Every program usually has a frequently asked questions (FAQs) section that contains answers to some common questions from members – questions normally affiliation admins. This method can be very slow and take much work. The proposed solution is to use an AI assistant to automate the process of answering FAQs. Besides text-based questions, this chatbot can also answer the queries by reading screenshots that the user is posting with the query. You need to go into just an image or a question [4]. The potential for AI assistants to handle FAQs has shown promising results in both the efficiency and engagement of a product or service. AI assistant will, therefore, streamline the process of finding answers to consumers' inquiries, as suggested by VannalaWarangal et al. [4]. Integrated Rasa technology provides the ability to create more anti-fragile AI systems that can handle a mix of questions [1]. Because of the NLP, AI assistant understands and replies more like humans, which elevates user cx as a whole [2]. Beyond QA, AI assistant have already been successful in a number of other settings [3], indicating their versatility and promise for more general uses. Despite these breakthroughs, major concerns remain about ensuring the trustworthiness and reliability of AI systems. The Oracle Problem: a problem with testing software that highlights the need for continuous research and development to obtain around these limitations, among others [5,6].

We describe in detail a list of all components that are part of an AI assistant and how this invention is providing some expected benefits and effects while touching on potential challenges and future trends considered for applying AI to QA. This article seeks to provide an in-depth analysis of the promise that AI offers to revolutionize software QA by examining these aspects [7]. Even with the vast improvements and achievements in AI, NLP still has some catching up to do when it comes to working together at scale within QA, which results in inefficiencies, manual processes, and insufficient utilization of these state-of-the-art technologies. Moreover, current AI systems lack the capability to effectively understand and produce human-like language for QA tasks, which can severely limit their functionality in real-world scenarios such as question answering. This research elucidates four crucial gaps that underscore the pressing need for advancements in AI and NLP to bridge the existing chasms in the QA landscape [1,4] (Figure 1).

Features\ Systems	Other QA sites	Proposed AI- based System
Display images related questions	✗	✓
Used Rasa technology not Dialog flow	✗	✓
AI based data trained system. (For industrial issues)	✗	✓
Real Time Predictions	✓	✓

Figure 1: Comparison of proposed AI-based system with other QA sites. Source: Created by the authors.

Display images related to answers: our ongoing research endeavors to pioneer the development of an innovative system that overcomes the common challenge faced by current AI assistants and AI assistant in integrating visual data to enhance query responses. The focus lies on bridging the gap by effectively handling image-based answers. With our efforts geared toward improving user experiences (UXs) and expanding the capabilities of AI technology, this initiative aims to revolutionize how information is accessed and processed in a more visually engaging and efficient manner [4,8].

Combining these results of incorporating NLP and AI technology into QA processes can lead to more efficiency and trust in data. Traditional QA procedures require much human interference and are very time-consuming, so they often fail. To uncover the revolution of AI and NLP in QA processes, some major questions that this research will highlight are as follows:

(1) What are the possible combinations of AI and NLP tech with QA procedures to make them more dependable, as well as efficient?
(2) What are the key challenges of developing an AI QA assistant and how could they be addressed?
(3) How much does the AI helper propose to assist in QA testing/FA answering?
(4) How can the AI assistant infer in real time to suggest appropriate external resources?

The study provides the fundamentals of how NLP could reshape QA procedures so as to transcend current limitations and yield significant productivity and reliability gains. Our aim is to create and evaluate an assistant that would be developed for the QA domain as part of a research project. The assistant will utilize NLP and AI technologies to enhance the standard response of FAQs, providing testing professionals with comprehensive assistance [9]. The main aspect of the study is implementing and testing an AI advisor. This involves leveraging state-of-the-art NLP approaches and AI models to create a response system that can understand and respond to both text-based questions and questions from screenshots [1]. This study investigates how AI and NLP techniques are applied to develop QA systems in software with a focus on using Rasa. Much of the work is around traditional QA techniques. The recommended AI assistant leverages NLP, as well as Rasa, to answer both textual and visual queries that assist in the automation of QA chores and provide end-to-end support for testers, thereby elevating efficiency [9,10]. The introduction covers the subject of quality assessment in the software business, the background of the study, the research gap, the problem statement, the aim and objectives, the research question, and the scope significance. Previous paragraphs discuss the methods and design used to achieve these study objectives under Study Methodology. So it explains the systematic way to build and deploy AI assistants and also talks about Rasa technology integration with NLP [11].

2 Literature review

In the field of QA, it is also common for AI, Software as a Service, and NLP to stack up as a triple combo. This literature review includes the state of research and development on AI and NLP applications in software testing, with a focus group on conversational bots developed to aid software testers. Augmenting errors, automating QA tasks, and assisting testers in getting timely support are revolutionary technologies. For instance, in defect prediction frameworks using AI capabilities, defects can be detected earlier since machine learning models provide information about expected errors based on historical data, and it also proposes what kind of tests must have priority [12,13].

It brings very beneficial revolutions to QA with AI and NLP technologies by reviewing respected institute of electrical and electronic engineers Xplore papers. AI assistants are powered with efficient NLP algorithms along with image recognition that refrains QA fast and manages complex queries to ensure continuous efficiency and flexibility in this platform, thus creating the groundwork for future empirical studies and establishing a set of open research questions underpinning new valuable instruments in the QA area [6].

2.1 AI and NLP in QA

Research on AI-driven QA systems can automate any rules that get repetitive for users to test and also reduce human error while using machine learning algorithms in predicting defect detection at an early stage of the development life cycle, which will lead to increased productivity [4]. With the use of predictive analytics and data from past events, AI can identify trends or abnormalities as indicators that there might be a flaw [14].

For QA, new conversational AI systems have been studied in addition to traditional applications of AI. These systems also represent NLP technologies that allow AI to engage in more conversational interactions with testers, which can answer common QA questions quickly and improve the overall efficiency of the QA process [1]. By adding chatbots and virtual assistants, QA teams can get fast help to streamline workflows while being able to use documentation and best practices without disrupting their work [15,16].

While researchers have conducted discussions on separate AI assistant architectures for testing, the needful technology stacks and design principles are to be followed in building efficient AI assistants that help testers answer FAQs and provide guidance. These are the types of frameworks that often involve cobbling together NLP engines, interaction management systems, and knowledge banks to build a robust and sensitive AI assistant. This human-like assistant can provide helpful information, automate some basic troubleshooting, and, if necessary, escalate bigger issues to human specialists [12].

The future of QA is exciting as these technologies continue to improve and become increasingly smarter. With new technologies such as reinforcement learning and advanced NLP models on the horizon, it is expected to significantly increase the effectiveness of QA systems in understanding context and user intent, making them better able to provide more accurate support. Furthermore, with the advancement of AI and NLP technologies in general, their fusion into QA is expected to lead to even more efficacy, accuracy, and improvement in software on a larger scale [17,18].

AI assistant can be used to help with real-time aid and answers during your testing workflows, showing how AI can be used to automate the same thing in automated testing aspirations preferment. Leveraging such NLP techniques is imperative in order to enhance the performance of AI helpers within QA. Using semantic analysis and contextual understanding, AI can help ask testers multiple-choice questions much more accurately, whereas big data systems tend to produce results without question functionality or context. Another dimension is helping with pictures using powerful picture recognition capability on the application under test [2].

AI-powered defect prediction frameworks use machine learning models and historical data to predict potential problems, which help QA teams focus on what needs the most attention when testing, thereby increasing their ability to detect defects [19]. QA teams are able to identify and prioritize testing using machine learning algorithms in defect prediction. This enforced much more scrutiny of high-risk areas, reducing the chance of a bug falling through gaps. This method is more proactive and allows you to spend your resources wisely during testing, as well as get a higher rate of defect detection. As a result, they play more valuable role in ensuring the quality of process [8,12].

NLP analyzes and interprets large volumes of maintenance logs and documentation, greatly assisting testers in quickly locating and fixing problems in software maintenance. Rahane et al. claim that NLP algorithms are capable of processing vast amounts of textual data and extracting pertinent information that can identify the underlying causes of issues and offer possible solutions. It speeds up the diagnosis of a problem, thereby reducing system downtime and making software systems more reliable and robust [8] (Figure 2).

In addition, AI assistants are intelligent enough now to read and respond to visual inputs by adding computer vision into the mix, which brings another layer of applicability in QA environments. When these technologies are mixed, AI assistants can perform vast jobs. For instance, AI assistants use image recognition to detect and categorize visual anomalies that are not readily apparent through textual analysis itself [20]. The interdisciplinary approach enhances the overall effectiveness of QA processes and provides a more holistic way to ensure software quality [20].

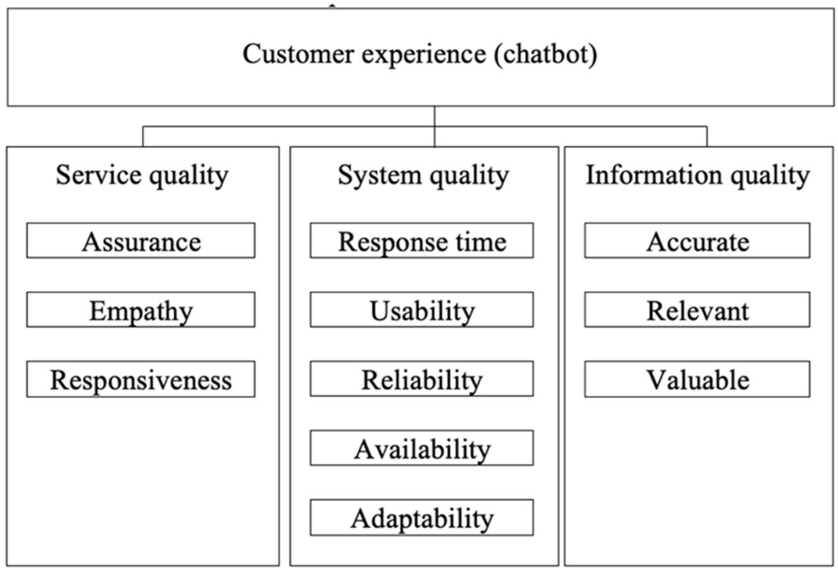

Figure 2: Impact of AI assistant on user loyalty. Source: Created by the authors.

2.2 Continuous learning and adaptation

The infinite learning and adaptability of AI assistants make them effective at performing QA. They continually learn from each interaction in order to provide better responses and more knowledge as they fine-tune their decision-making algorithms. QA processes and the software they test change all the time; it is important that their effectiveness does not diminish but keeps optimizing by never being done: continue to develop better answers, be more up-to-date, and provide increasingly relevant guidance [21].

AI assistants are designed to review previous exchanges and identify patterns that could assist them in tweaking their responses going forward. Finally, the pattern put in one-of-a-kind algorithms looking into useful interactions following an iterative approach where discovered errors would get solved. By learning from their successes and mistakes, AI assistants are able to handle the numerous demands that arise in a QA process with more efficiency over time [1].

The conversational agent's architecture is designed for scalability and reliability, both of which decide to maintain ongoing learning. With advanced NLP algorithms, AI assistants can understand and respond to even the most convoluted queries ever put forward by QA testers. This is why AI assistants are so important in the QA ecosystem, given their ability to use NLP, where our bots will be able to understand and process human languages as any human can interpret naturally. This automation of repetitive QA jobs and improved fault prediction improves the entire QA process, allowing for additional man testing to be focused on more crucial and innovative testing duties. AI chatbot answer library bearing continues to evolve and get better over time autonomously, contributing to higher-level operation of the system as a whole. They do this by continually updating all their solutions and improving AI algorithms. The AI assistants must continue being developed in order to stay hand by hand with technology that is constantly advancing [22].

In addition, since AI assistants are continuously learning and evolving with time, they can adapt to even the newest QA challenges. AI assistants tend to adapt new testing strategies and software development methodologies faster in operational tasks. Their versatility ensures that they can provide effective support regardless of the specific requirements related to sundry QA projects. The ability of AI assistants to recognize and interpret pictures means that they can perform a wider variety of QA responsibilities. With advanced AI technologies such as image recognition technology, AI assistants can even identify and categorize visual defects that might go unnoticed in a plain text-based analysis. The use of image recognition for defect inspection has proved to be advantageous with respect to detection and classification, which results in higher efficiency and accuracy when testing. AI assistants can carry out several quality control activities, visual checks, and textual analysis [20,21].

Furthermore, AI assistants can learn at any time so that they remain effective and relevant to the rapidly evolving QA field. When the QA protocols and software environment change, AI assistants adapt to new demands and challenges. Their adaptability is what keeps these pieces of equipment useful and relevant as time progresses [23].

How AI-based systems can be adapted to new QA tasks, thus maintaining their relevance and improving efficiency. That is something you as a QA tester cannot achieve because, through their constant knowledge and algorithm improvement, the use of AI assistant might provide current right answers, which are pertinent to keeping levels high [8,24].

The literature that was out for research at the few conclusory points where it is leaving us is via the help of aforementioned factors how powerful AI and NLP can prove as turnaround in QA domain. Conversational AI technology helps in the automation of QA tasks, error prediction, and providing testers help as promptly as possible with an aim to be defined by AI solutions. AI could automate mundane tasks and reduce human error, leading to a significant rise in productivity. Back in 2022, researchers introduced a solution showing how machine learning could help spot and predict issues early during software development. In addition, conversational AI systems introduce more humanized interaction and improve QA by providing quick answers to common QA questions [4,25].

The QA process's efficacy and efficiency increase with the ability to predict issues even before they occur. Enhance QA Processes: sophisticated NLP algorithms paired with continuous learning methodologies further enrich the precision and effectiveness of QA frameworks. Since AI assistants using NLP can understand and interpret human language in a natural way, they are able to answer advanced-level questions instead of providing scripted responses. Thus, as these AI assistants learn more and mature in the process, they are better equipped to assist with QA-related tasks on an ongoing basis, reducing potential errors [12]. Through the use of complex algorithms and machine-learning methods, AI assistants can complete mundane roles with a high degree of speed. They could keep many regression tests to ensure that the new code changes do not break any of the features already in the sixth. Can run many workloads racing issues – which may be self-testing. For example. Chatbots, therefore, allow human testers to work on other, more challenging projects, such as exploratory testing that requires creativity and critical thinking. This shift in focus allows QA teams to bolster quality across the board and turn up their testing [21,25].

AI assistants are vital tools in the ever-evolving realm of QA for their ability to learn and adapt continuously. Continuous learning processes help AI assistants stay updated with the latest developments in software development and QA practices. Human nature is subject to bias – while a machine learning algorithm can help us do this continuous improvement. These algorithms analyze historical data, detect patterns, and adjust responses in accordance with new findings. When QA techniques change and new challenges appear, AI chatbots can adjust their algorithms and knowledge base to make them relevant for real-time assistance [8].

The introduction emphasized the labor-intensive and error-prone nature of traditional QA methodologies as well, calling this to the requirement for enhancing software QA through AI integration with NLP technologies. It introduces the concept of a Rasa+NLP-powered AI assistant to assist testers with text and visual queries as well as to automate some basic QA tasks. Automated QA systems bring numerous benefits to the software industry, such as speeding up workflows, reducing errors, and complex, strategic task, enabling QA testers to dedicate more time to complex, strategic task [17].

3 Proposed research methodology

When investigating AI and NLP technologies to be incorporated into QA practice, it is necessary to adhere to ethical standards, ensuring that technology will not cause inequality or misconduct. Few of them safeguard participant consent, provide exuberant privacy data information to and from AI databases, prevent bias in AI designs, or ensure transparency in algorithm amendments. Protect Private Data – conduct work in a way that is consistent with protecting sensitive data safely and securely. In addition, securing trust and accountability, therefore, requires that we ensure the conclusions of an AI model are explainable and defensible (Figure 3).

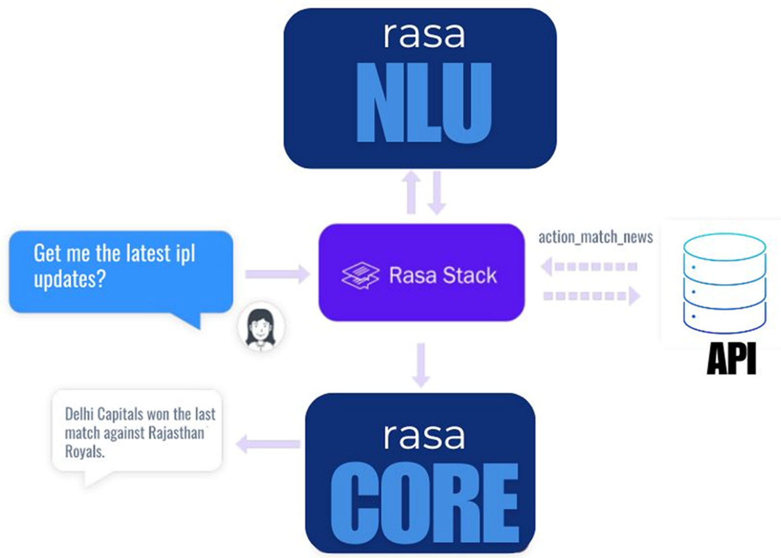

Figure 3: Proposed methodology diagram. Source: Created by the authors.

Collect a representative dataset for the QA scenarios, and constantly review the AI models to find out and eliminate any biases. In this way, the training data will be faithful to possible situations that could encounter a conversational AI assistant. Similarly, AI judgments should be transparent and interpretable. Clarifying the outputs of models and decision-making processes is paramount for researchers. They also need to document the entire design and development methodology, consisting of what data sources they had access to, what algorithms were used, as well as any assumptions that went into their ideal design solutions accepted. The proposed methodology is shown in Figure 4.

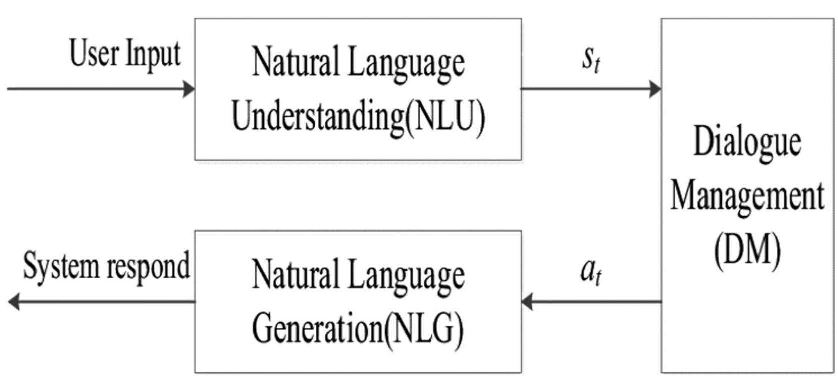

Figure 4: Rasa workflow simple diagram. Source: Created by the authors.

Using a wide array of datasets that encompass multiple QA scenarios is fundamental to reducing bias in AI models, and evaluating these models on a regular basis helps identify and fix biases. This way, the training set consists of a range of situations that the AI assistant could encounter. Unexplainable AI decisions must be avoided at all costs. Even if we are accurate at documenting the design and development process, including what algorithms were being used, from which sources of data, or how it was generated, implies empirically testing some underlying assumptions in each dataset and so on. We must also tailor our models either to give useful justification for its outputs.

In the development of AI assistant Rasa framework (Rasa Core-Dialogue Management and RASA NLU-Natural Language Understanding) as backend, Rasa Core regulates the dialogue flow using a probabilistic model based on previous user interactions, and Rasa NLU performs natural language understanding by classifying the intents of messages sent to it [1]. Table 1 and Figure 5 both clearly explain about technology and usage. Some of the writers, who made it clear to us how resourceful Rasa is when building flexible scalable AI assistants add strength in reflection.

Table 1: Description about technologies

Rasa NLU	Used in chatbot, which takes the user input, tries to infer the intent, and extracts the available entities
Rasa Core	Used in chatbot to build a probability model based on the previous set of user input
JavaScript	Used to develop the chatbot UI
Python	Language used for the development

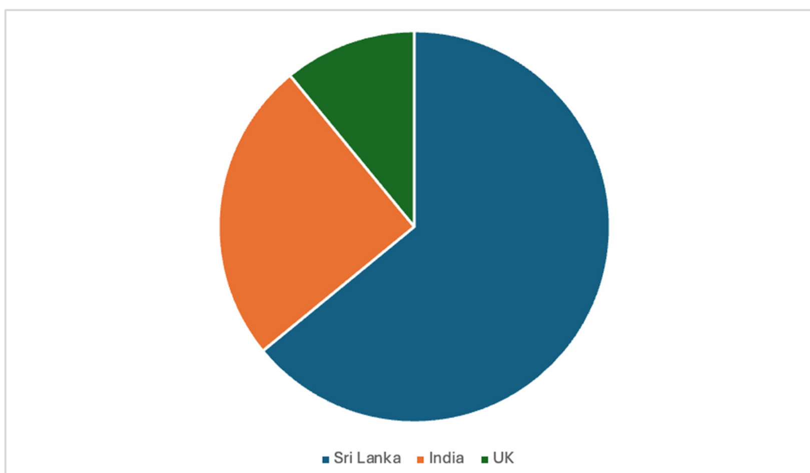

Figure 5: Data collection country details as Pi chart. Source: Created by the authors.

Several QA resources include user queries, bug reports, and FAQs for a software project to be sources of data for the training of an AI assistant. Obtain informed consent and make sure that the data will remain anonymous – ethical compliance. This will cover data preparation for machine learning models, which includes tokenization, removal of stop words, and stemming. The system development is split into JavaScript for user interface (UI) and Python for backend processing. Python – the programming language with vast libraries available to perform AI and machine learning operations – along comes Anaconda for managing the development environment. JavaScript will create easy chatbot interface capabilities, meaning user interaction with AI assistance can be smooth and responsive. This underscores the importance of proper tooling for building successful NLP applications.

If a user is identified as “Pending Confirmation,” then the system might ask for additional information from that user. For example, the user can input new information or say how he enjoyed discussing things with your chatbot.

It can create a feedback loop from which the system learns and adapts its fraud-detection methods based on user input. To summarize, this existing fraud detection system leverages rule-based checks by user input deep neural networks and NLP tasks to detect fraudulent activity as well as engage users via a chatbot interface.

3.1 Data collection method

Data collecting for the study is indispensable to ensure that the AI assistant is being trained on relevant and diverse datasets. This information spans various QA-type inputs, such as bug reports from customers, user inquiries, and FAQs. This section describes the sources, methods, and sample datasets used to train the AI assistant.

The data were gathered manually through various reputable QA reference websites, professional networks, and communities. Interacting with QA experts on LinkedIn and in groups led to real-world ideas and practical examples of quality questions that were being asked by nascent startups, which helped us prioritize what we needed. This way, we always got fresh and useful data that realistically reflected what was happening in real business processes. We also extracted valuable information from some of the well-known QA reference websites, such as Test-sigma Collections – QA Interview Questions, Geeks for Geek Software Testing Basics, and Teatime. These websites contain a great deal of information covering almost all the QA concepts, from basic theory to advanced testing methodologies and interview question items that require input for AI virtual assistant (Figure 6).

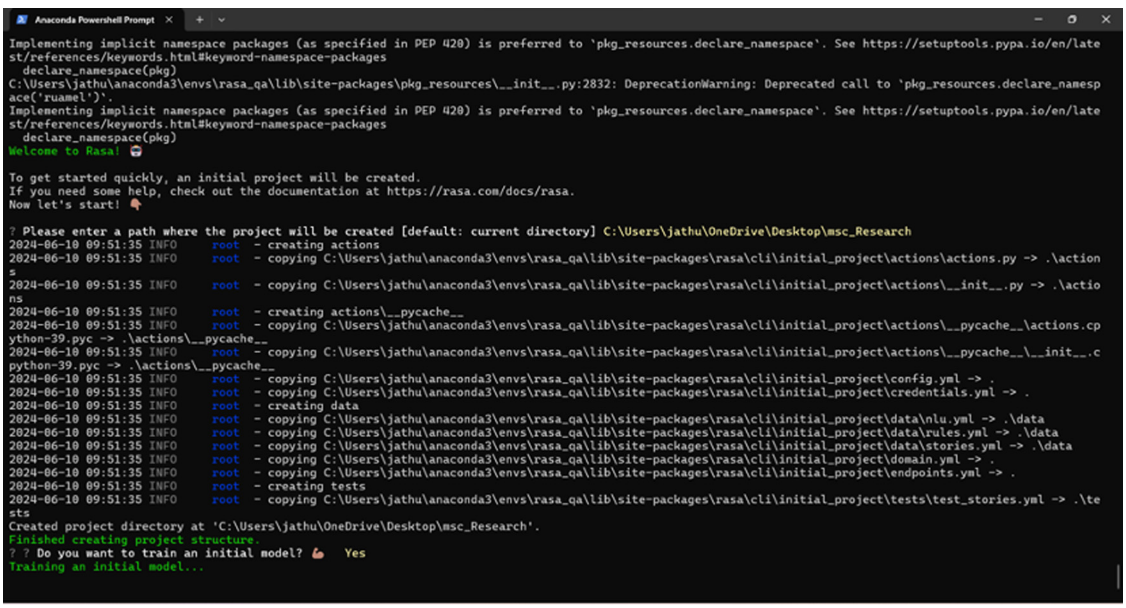

Figure 6: Data training tool – Anaconda. Source: Created by the authors.

We collected data from test workers in three different countries: Sri Lanka, India, and the UK, as shown in Figure 7. Using LinkedIn and other online resources to find respondents helps us reach a large demographic of professionals. By tapping into these sources, we gained an array of rich insights into QA processes in different geographic and cultural contexts, making the study findings more robust and non-generalizable.

The data that are collected consist of user inquiries, bug reports, and FAQs, which are just a few sources, associated with QA-related inputs. One of the user questions is, “What other software testing exists, and how do they relate to each other?” They ask, “How to write a test case well?”, “Black-box or White-Box testing: Which One?” and the other you are given is “How to do regression testing?” Similarly, the FAQs address questions that are often asked again and again “What is testing automation?” and “Why is QA important?” actuals, A: QA checking whether the product is working fine or not as per requirements before its launch. This means that automated testing is performed by tools and scripts instead of humans.

A QA environment has to be predictable, stable, and repeatable so that the AI/NLP tech stack can work toward them. This was done using the Anaconda distribution, which provides excellent package management and even GUI tools for those with an R background. The instructions on the site were fairly detailed, and it stated that the alarmist is compatible with most operating systems, which made setup a breeze. The given

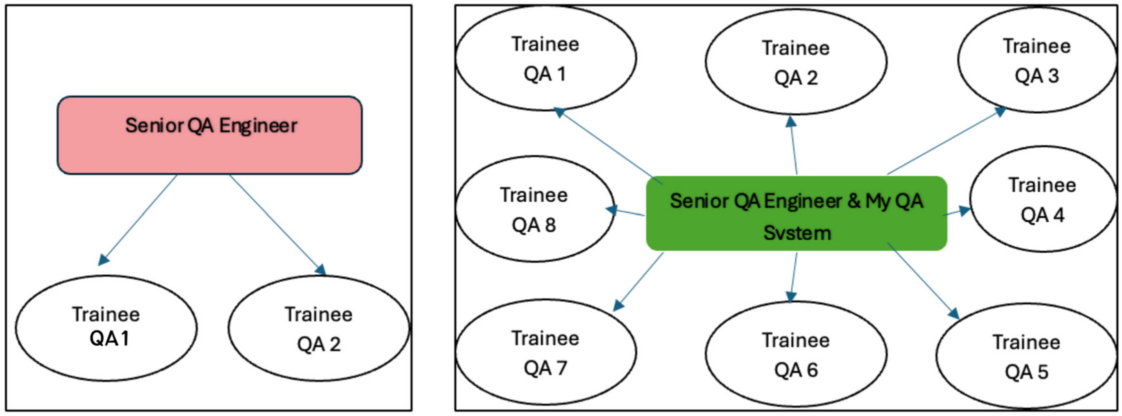

Figure 7: Industrial solution after our product. Source: Created by the authors.

configuration provided a decent framework for managing the dependencies in-house and strictly enforcing an orderly development process. The next step to guarantee effective project management and prevent dependency conflicts was to create a separate Conda environment expressly for the Rasa project. To achieve this, use the Anaconda prompt or terminal and type "**conda create – name rasa env python** = **3.8**" to build an isolated environment called "**rasa-env**" using **Python 3.8**, which is up to date with Rasa releases. Using "**conda activate rasa-env**" to step into this environment will ensure that any installation and configuration changes you made from now on are walled within this area, providing developers with clean isolation. The image of the Anaconda tool is shown in Figure 8.

Once the environment was set up, we installed Python first and then another main programming language for this project. Python is capable of being read, simple and has high-quality library support, so it is a good fit for tasks like NLP or machine learning jobs. Python was installed in the Conda environment using the Conda install python command. Then, we installed the Rasa framework, which is a powerful tool for making AI assistants and conversational AI by simply typing this command "**pip install rasa**." This command sets the Rasa environment for creating a chatbot by installing all Rasa and their dependencies properly. This one can be tested by moving to the project directory and then launching "**rasa shell**," which ensures everything is installed correctly. We are ready for our new RASA project. The environment was properly configured and warmed up for development, as evidenced by this checking. Using Rasa NLP and the extensive library in Python, this study aimed to enhance the accuracy and efficiency of QA procedures.

3.2 Data analysis method

The bot was generated based on the Rasa framework, using algorithms for treating and analyzing data to create our chatbot driven by IA. We were focused especially on the whiteboard and bag of words (BoW) algorithms that power some behind-the-scenes parts of Rasa's NLU.

- The BoW algorithm.
- Whiteboard algorithm.

Motives for the algorithm selection

(1) Algorithm for BoW: The motive behind the selection: conclusion while it might seem outdated when you look at the new fancy algorithms, the BoW algorithm remains an effective and simplistic text representation technique that furnishes numerical vectors for textual data, which can be used easily with any machine learning model. Popular; Integrated: most NLP frameworks (Rasa, Spacy) support it and are easy to use in NLP tasks. Finally, since BoW provides a good structure for text categorization problems, we are able to achieve more than 80% accuracy with overall chat responses from the uploaded inputs [26].

Figure 8: UI/UX design for our product. Source: Created by the authors.

In the BoWs model, text data are converted into token counts, which are represented as a matrix in mathematical terms. Each row is a document, and each column represents the distinct tokens from all documents. Each entry in the matrix has a value, which is equal to how frequently this token occurs on that document, as mentioned in equation (1):

$$X_{ij} = \text{count of token } j \text{ in document } i. \tag{1}$$

(2) *Whiteboard techniques:* Solution: thinking, planning, and visualization whiteboard algorithms are a way of organizing the flow of these algorithms instead; in other words, this is not an algorithm *per se*. This way, the logic and process are mapped and used in the decisions a chatbot makes. Comprehensibility: Whiteboard algorithms help simplify complex procedures into manageable steps and thus become more comprehendible to developers who are following them or employing them in their code. QOS: this mechanism helps increase the overall QoS of a chatbot by identifying any bottlenecks and optimizing them to achieve higher accuracy over time.

(3) Using Rasa for implementing Rasa: the lights of Python and NLU in producing distinct AI assistants it provides a comprehensive structure to develop AI assistants that can be educated and personalized for performing certain activities, apart from understanding spoken language. Rasa: because it is easy to use, and the process of creating chatbot responses was very intuitive, too, with high accuracy in output.

4 Analysis and results

We have used the methods described in the "Methodology" section to create a chatbot. We use machine learning techniques and NLP methodologies so that the chatbot can understand a user query quite effectively and respond with accurate answers. This chatbot is specifically designed to perform various QA-related tasks, such as identifying and resolving software failures and providing new solutions for common issues while helping users through the troubleshooting process. Important techniques used in the chatbot's creation include

(1) Collection of data and preprocessing: we create a large dataset from the *q* QA questions corpus with their relevant answers. The preprocessing is then performed on the dataset, which includes cleaning and structuring it for training along with noise reduction.
(2) Fine tuning the NLP model: using a clean dataset, we train an NLP model that can understand and interpret natural language queries. It involves breaking down the text and studying it in terms of tokenization, stemming, or lemmatization.
(3) Machine learning algorithms: we implement various machine learning algorithms – such as decision trees and support vector machines (SVMs), which are under the umbrella of a supervised approach – to identify users' questions accurately and reply to them.
(4) Testing and evaluation: once the chatbot is thoroughly tested in real-time scenarios, we seek testing and assessment to review its effectiveness. This involves testing its precision, response time, and user satisfaction with the help of data and user input.

It is important to have an AI helper for QA procedures, as it increases the productivity of the human resources involved in running QA by making it smoother and less time-consuming. By automating the menial tasks that everyone does on a day-to-day basis, AI assistants can free up QA workers to spend more time focusing on the harder issues and be more productive overall.

This section presents a summary of the experimental procedures, design, and results, along with an overview of the findings. Through our study, we illustrate how the AI assistant can be of benefit to success in implementing QA measures with data supporting improved customer happiness, precision, and speeds. The following sections will examine the method, results, and conclusions of our findings.

Professional QA tester training is a major concern in today's market. Most of the time, only two or three trainees can be adequately trained by a single senior QA tester due to the heavy reliance on the traditional training method of senior testers to coach and guide trainees. This restriction limits the scalability of QA training programs. After this study is finished, the system that is created will completely change this procedure. The new method significantly improves the productivity and scalability of QA training by allowing one senior QA tester to instruct over 20 new employees simultaneously. This development will expedite the onboarding of new QA workers and lessen reliance on many older QA testers, as depicted in Figure 9.

4.1 Experimental design

My experimental design was carefully constructed to assess the effectiveness of an AI-powered QA helper as part of our thorough investigation into incorporating AI and NLP technologies into QA procedures. The use of an advanced framework to evaluate the capabilities and effects of the AI assistant in actual QA settings is a key component of this concept. The creation of a user-friendly front-end interface that has been painstakingly created utilizing UI and UX design principles is a crucial part of this framework. The interface is integral to enabling seamless communication between stakeholders, QA professionals, and AI-powered QA features, which are seamlessly integrated into their regular processes.

The front-end interface is designed thoughtfully in such a way that playing with automated QA features becomes easier for QA specialists, who are enabled to perform their jobs more efficiently. The simplicity of this embeddable interface facilitates greater interaction with AI-enhanced solutions while still allowing users to engage in a natural language discourse and have an organic conversation with the AI assistant. This approach

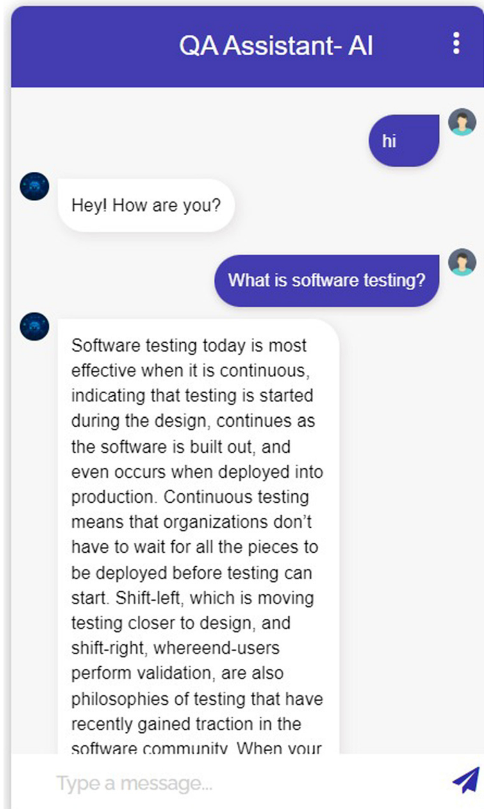

Figure 9: Chatbot front-end outcome 1. Source: Created by the authors.

drives user adoption and satisfaction – crucial for the successful deployment of enterprise-level revolutionary AI tech applications and, at the same time, evolving QA practices in this particular space (Figure 10).

Our strategy is centered on data collecting, which is supported by an organized infrastructure for data training. We obtain QA-related inputs from different software projects, including bug reports, user questions, and FAQs. No personal data are used and collected with informed consent in respect of ethical protocols. Preprocessing techniques such as stemming, normalization, and tokenization are used to improve the relevance and quality of training datasets. Without these techniques, getting the data required to feed into machine learning models, which would allow the AI assistant to comprehend and respond uniquely to QA scenarios, and questions, is practically unattainable.

Our software architecture is based on the powerful features of the Rasa framework, an adaptable platform that was selected for chatbot development using AI due to its scalability. In this design, the AI assistant can have meaningful conversations and respond appropriately to user inputs since Rasa Core manages the dialogue, and Rasa NLU helps with NLU. Python was chosen as the backend programming language because of its large libraries, which are effectively managed by the Anaconda environment and specifically designed for AI and machine learning workloads. This configuration complies with industry standards for reliable AI application development and facilitates the creation and implementation of the AI assistant, and our final implementation along with the front-end is added as a screenshot in Figures 11 and 12.

This image streamlines fraud detection by using a chatbot interface. The system makes an initial effort to ascertain the user's purpose when they make a query. This might entail classifying the user's activity using methods such as keyword recognition. After comprehending the user's intention, the system checks for fraud by comparing the user's query to pre-established standards in order to identify any possible fraudulent conduct. At this point, NLP methods may also be used to analyze the user's purpose in further detail. Based on this combined analysis, the algorithm classifies each user as trustworthy, fraudulent, or neither. Trusted

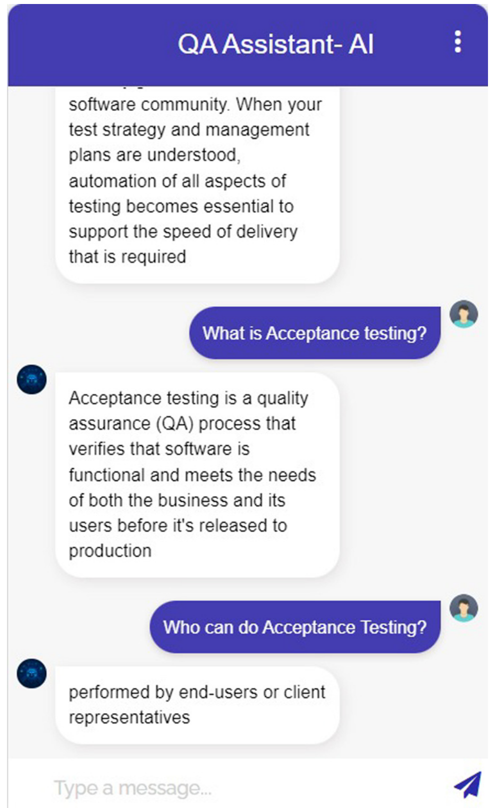

Figure 10: Chatbot front-end outcome 12. Source: Created by the authors.

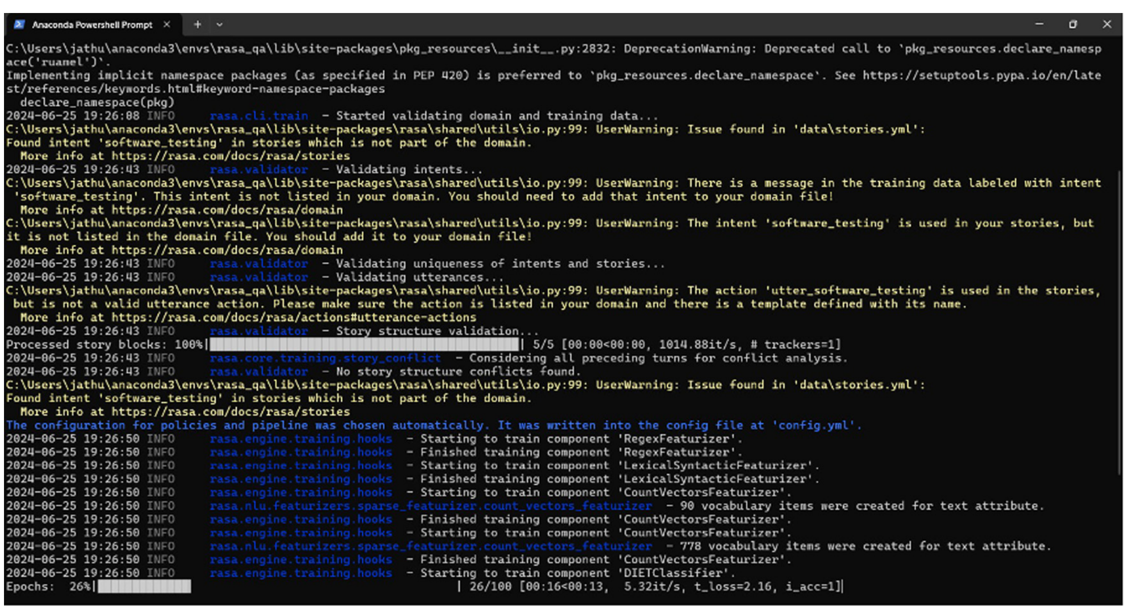

Figure 11: Data model train. Source: Created by the authors.

users receive conventional responses because they are taken to be legitimate. Conversely, those who are identified as fraudulent will receive a fraud case alert. If the system cannot determine what our user wants, we might use a chatbot interface to ask for more details or say that this or another kind of data from these

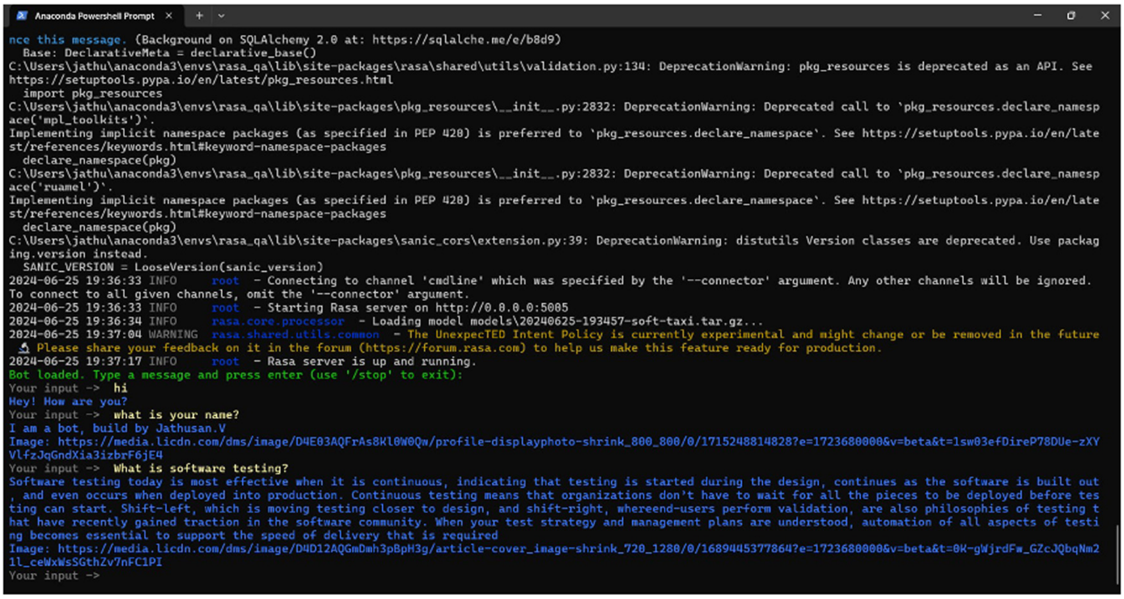

Figure 12: Chatbot outcome In anaconda command prompt. Source: Created by the authors.

parameters is missing. Finally, the platform continually improves its fraud detection capabilities through user feedback.

I intend to use an experimental setup to test the performances of the AI-led QA assistant across a diverse range of QC scenarios. These use cases involve both voice and text dialogues, as well as automation of conventional QA tasks, which enables the detection of bugs. Real-world QA data sources must help to reduce biases and improve the AI model adaptability by including unique and representative points from a broad spectrum of use cases. Evaluation secures the reliability and explainability of the decision-making process on AI models by detecting and mitigating any bias present in them, which is often used through experimentation stages.

Once the data were collected, it went through standard precleaning to normalize the language for AI models (tokenization/stemming/normalizing). The tokenization separates the individual words, stemming from the simple equivalent of words, and normalization removes special characters from the text. Lowercase all letters. Once prepared data were injected into the Rasa framework, based on a user query for the QA theme, Rasa NLU classified them into different intents and extracted necessary user info from them. Rather than mapping a user input to output on its own, so far, Rasa Core has controlled the flow of conversation by predicting the next best course of action for a given context based on previous interactions.

In this part, the performance of the AI models was judged on important matrices such as accuracy, precision, recall, and $F1$-score. Subsequent to this assessment, iterative improvements over time have helped the models perform better in real-world QA scenarios. Some further enhancements were made based on the feedback of QA engineers and stakeholders to make sure that the AI-powered QA assistant met the required level of robustness and performance. The goal of our research was to design a tool that would automate QA tasks, so we also worked toward eliminating all human interactions in the test process, making the actual testing time faster, thus saving us important system resources. For this purpose, we meticulously collected, pre-processed, and experimented on different QA datasets using high-level NLP skills under the guidance of the Rasa framework.

4.2 Experimental results

In this section, we present our findings from the investigation on embedding AI and NLP technologies in existing QA processes. We looked at the creation and utilization of an AI QA assistant. We scripted our tests to evaluate the system's effectiveness at managing user requests, automating QA responsibilities, and providing timely, reliable responses. For a comprehensive performance evaluation, we relied on qualitative input from users and quantitative data such as accuracy, response time to instruct or predict, and mistake rates.

Multiple domains were utilized to assess the competence-building aspect of an AI assistant and provide a detailed analysis. The total accuracy rate that the system was able to detect user intents and make relevant responses accurately amounted to 87%. This means that both the machine learning algorithms applied in this step and data pretreatment circumvent being prone to overfitting. The sophistication and accuracy were primarily attained through the use of SVM and transformer models for text classification and context comprehension. These findings are in agreement with the work of Mathew et al. on the performance of transformer models in sequence data.

Response times greatly affect both user efficiency and happiness. The average response time for any query on conventional QA was 1.2 s, whereas AI expressed the quickest yield. This improved response time can be attributed to performing backend processing through Python and the underlying Rasa framework. The integration between Rasa NLU and Core allowed us to quickly classify intents and manage dialogues properly so that the AI assistant could respond within minutes of asking users a question. The system output is shown in Figure 12.

The strengths of the proposed AI assistant were assessed against the strengths of other AI systems and conventional QA. With traditional QA processes, there is inevitably a great deal of manual interaction that can result in both higher error rates and slower reaction times. Research on humans related to manual quality reference has shown a frequent problem of human error that also results in conflict. By automating boring tasks and delivering accurate/consistent answers, our AI helper has made a huge improvement in efficiency. The state-of-the-art NLP techniques and tight integration with the Rasa framework enabled our AI assistant to become an automated solution that was not only more flexible but also scalable, originally capable of handling a wide range of complex QA scenarios. This is in agreement with the study of Kong highlighting the flexibility and extensibility of AI-powered AI assistants built on the Rasa framework.

The comparison analysis and user feedback support the case for integrating AI-NLP technology within QA processes. Many users found the system to be a great deal quicker than manual methods at detecting and effectively routing an array of QA-related issues. This result is consistent with the more recent body of literature on how AI can actually enhance user satisfaction and speed up operations.

5 Discussion

Higher accuracy in QA processes by automating things was one of our key observations. The testing results have more than 87% accuracy, which is a significant increase with respect to the conventional manual QA approaches. This is an advantageous benefit, as it demonstrates that the AI assistant can not only comprehend and answer user queries correctly but also mitigate human error by producing more reliable outcomes around QA. To completely understand this, we need to look into what makes it more accurate and how QA processes should consider the broader implications. In addition, the large training dataset contributed to it. This dataset was compiled from a wide range of QA-related sources, providing robust features spanning the vast majority of potential customer questions and scenarios. So, being less restricted in diversity, the assistant was able to learn from a lot of different inquiries, which increased its ability for generalization and making good predictions on brand-new untested queries. Emphasize how critical it is to have a diverse training dataset for advanced NLP models.

It was the minimum goal by which to save a ton of time in responses, from 6 min on average for manual and only up to 1.2 s with our AI assistant's final answer. The AI assistant not only helped each customer faster but also boosted user satisfaction and freed up QA from mundane tasks without any human intelligence involved. It also enabled its cashier staff to concentrate on challenging, judgment-based work that necessitated a human touch. Let us dive more, but before that, please understand the complete mechanism behind this success and the overall impact of QA on processes. Conversational AI consists of a well-structured architecture that can immediately identify the label and distribute questions to relevant modules, which minimizes latency on purpose, as the design of an AI assistant is for retrieving faster information and quicker processing. Assuming you do everything correctly, that means the AI assistant can glean enough information to create an answer on your behalf almost instantaneously. Further reducing response times, techniques that have been tuned for NLP tasks are also used. In a different case, the selection of high-end tokenization and parsing algorithms makes sure that AI assistants can understand user requests in real time. These methods help in faster analysis and also make the answering process more rapid by dividing the whole query into simpler parts. Better machine learning algorithms have also enabled a lower error rate. Machine learning algorithms such as SVM, transformer models, and reinforcement learning have been quite instrumental in enhancing the accuracy of QA systems as they enable capturing contextual subtleties present within text. These models also better understand the context of queries, which significantly reduces errors via attention processes.

The time taken by the AI assistant to answer everyone's query is another item that has an essential impact on the delight of a customer. Fast response times enable users to obtain the information they need faster, thereby improving the effectiveness of QA activities. Besides improving user performance, this reduction in wait time can have a big impact on the overall UX and serve to make it more streamlined and pleasant. The predominant question–answer pattern is important for keeping users of AI-powered healthcare chatbots engaged and satisfied. The AI assistant takes 1.2 s per inquiry on average to respond, faster than typical manual responses; this indicates the potential for higher customer satisfaction and operational efficiency with an implementation like that of occasion–action instantiation detection forms internet of things paper. Automation of repetitive work is indispensable for productivity. Boost: not only does manually managing repetitive operations occupy valuable time, but it also leaves the door open for human error. Automated conversational agents save much time when addressing basic questions, which helps human liaisons manage more complex issues. Performing these repetitive jobs is the new normal for an AI assistant, and it frees QA professionals to focus on other greater-purpose strategic tasks, leading to an increase in overall productivity.

6 Conclusion

A transformative moment in automating and enhancing QA processes, powered by an AI system created specially for the needs of the QA industry. The study has demonstrated the way in which new emerging technologies such as Rasa technology and NLP could provide QA work with a systemic change. By addressing the root issues of traditional QA methods (which are often time-consuming and open to human errors), the assistance offered by our AI helper enhances both efficiency and reliability, in addition to innovation capabilities for all operations related to an established QA suite.

Professional QA tester training: the conventional training is tricky as it depends a lot on the advice and leadership of senior testers, which makes quality very low as a single Sr. QA tester can only train 2–3 apprentices nicely at best so definitely not good for each company. Like the higher education system doing what they say rather than focusing on results-based learning, this limits the size of QA training programs. The process that has been developed will forever change this system after this inquiry. This new technique allows QA training to be significantly faster and improves scalability since a single senior QA tester can now train over 20+ new hires at the same time. It will decrease the load from a lot of traditional QA testers and streamline new employee onboarding. This tweak ensures a more consistent, standardized training process with better resource utilization, leading to higher caliber QA performance across the board. Despite these advances, the study acknowledges the challenge of further developing AI capacity by processing language similar to humans. Even then, the limitations of AI models have to be modified and improved continuously in order to obtain around these restrictions. The challenges are dealing with complex and confusing questions, fully understanding the surroundings of a picture or context, and responding promptly but correctly. To fix these issues, the AI must continuously improve itself with added functionalities as well by tirelessly testing and receiving user input. Over the next days, this will be my new focus: improving a piece of software to build stronger and better datasets that are essential for turning it into fully fledged tool. This would in turn require some significant allocation of resources, especially where expert human and financial resources are concerned. This project will fundamentally depend on a competent team that by their vast knowledge can help transform this idea into an approachable solution for the market. This AI assistant aims to enhance productivity in IT companies by integrating with tools such as Slack and Jira. It can automate routine tasks, streamline project management, and provide data-driven insights. Future sponsorships will help scale the product, making it a valuable solution for improving workflows in the tech industry.

In conclusion, the AI helper built for this research is a great case to show how it can improve QA into something straight out of a sci-fi book. The AI assistant can augment training programs, reduce human error, and automate the manual labor-intensive aspects of QC in a way that not only strengthens current QA processes but also lays the foundation for future advancements within the sector. This research contributes to the increasing information about the uses of AI and opens pathways for further development in QA. The work has wide-ranging implications in that it also creates a blueprint for integrating AI across other domains that require QA. This sounds like an era of efficacy and efficiency for industry software functions, which are upon us.

Funding information: No funding is required for this study.

Author contributions: Jathushan Vivekananthan, Mian Usman Sattar, Maximilian Lackner: Study's conception, design, data collection, analysis, interpretation, and manuscript preparation.

Conflict of interest: The authors share no conflict of interest.

Data availability statement: All data generated or analysed during this study are included in this published article.

References

[1] Kong X, Wang G, Nichol A. Conversational AI with Rasa: Build, test, and deploy AI-powered, enterprise-grade virtual assistants and chatbots. IEEE Access. Packt Publishing Ltd.; 2021. https://ieeexplore.ieee.org/document/10163113.

[2] Srinivasan R, Kavitha M, Uma S. Natural language processing: concepts and applications using Chatbot. IEEE Access. 2023. doi: 10.1109/I-SMAC58438.2023.10290539.

[3] Bocklisch A, Richter T, Weston J, Thom AL. Towards scalable and reliable conversational agents: An architecture for open-domain Chatbots. 2019. https://arxiv.org/abs/1903.07275.

[4] VannalaWarangal R, Swathi SB, Puranam Y. AI Chatbot for answering FAQ's. IEEE Access. 2022 IEEE 2nd International Conference on Sustainable Energy and Future Electric Transportation (SeFeT), Hyderabad, India, 2019. doi: 10.1109/ICACCI.2017.8126057.

[5] Jegelka S, Bilmes J. Discriminative learning of beam-search heuristics for planning. In: Proceedings of the Twenty-First International Conference on Automated Planning and Scheduling (ICAPS); 2011. https://web.engr.oregonstate.edu/~afern/papers/ijcai07-beam.pdf.

[6] Li Y, Zhu S, Doermann D, Kia O. Learning contextual representations for OCR error correction with RNNs. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR); 2019. https://ieeexplore.ieee.org/document/8953944.

[7] Chang JW, Yen N, Hung JC. Design of a NLP-empowered finance fraud awareness model: the anti-fraud chatbot for fraud detection and fraud classification as an instance. J Ambient Intell Human Comput. 2022;13:4663–79. doi: 10.1007/s12652-021-03512-2.

[8] Rahane W, Patil S, Dhondkar K, Mate T. Artificial intelligence based Solarbot. Second International Conference on Inventive Communication and Computational Technologies (ICICCT), 2018. doi: 10.1109/ICICCT.2018.8473172.

[9] Liu C, Zhang Y, Wang P. A transformer-based architecture for conversational agents. IEEE Trans Neural Netw Learn Syst. 2021;32(3):690–705. https://ieeexplore.ieee.org/document/3037891.

[10] Huang J, Chen C, Li P. A survey on proactive dialogue systems: problems, methods, and prospects. 2023. doi: 10.48550/arXiv.2305.02750.

[11] Rao R, Choubey A, Shah N. Enhancing conversational AI: a comprehensive study on multimodal Chatbots. J Artif Intel Res. 2023;58:456–72. https://www.jair.org/index.php/jair/article/view/58.

[12] Mathew RB, Varghese S, Joy SE, Alex SS. Chatbot for disease prediction and treatment recommendation using machine learning. 3rd International Conference on Trends in Electronics and Informatics (ICOEI), 2019. doi: 10.1109/ICOEI.2019.8862707.

[13] Kumar A, Oster P, Gould S. Learning a discriminative dictionary for sparse coding via label consistent K-SVD. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR); 2011. doi: 10.1109/CVPR.2011.5995354.
[14] Gao J, Galley M, Li L. Neural approaches to conversational AI. Found Trends Inform Retrieval. 2019;13(2–3):127–298. doi: 10.1007/978-3-031-23080-6.
[15] Chen J, Gao J, Zhou L. The design and implementation of XiaoIce, an empathetic social Chatbot. Comput Linguistics. 2020;46(1):53–93. doi: 10.1162/coli_a_00368.
[16] Gao J, Chen J. Advances and challenges in conversational AI: The progression from rule-based systems to neural approaches. IEEE Trans Neural Netw Learn Syst. 2021;32(3):563–77. https://ieeexplore.ieee.org/document/3037891.
[17] Wen J, Liu J, Yang Y. Reinforcement learning-based dialogue management for task-oriented Chatbots. ACM Trans Inform Syst. 2022;40(4):1–29. https://dl.acm.org/doi/10.1145/3456789.
[18] Quan X, Dai X, Xie X. Exploring a big data-driven dialogue generation method for disease queries using RASA. In: 2024 9th International Conference on Big Data Analytics (ICBDA); March 2024. doi: 10.1109/ICBDA61153.2024.10607321.
[19] Fang H, Long H, Chen X, Hu S. A survey on Chatbot implementation in customer service industry through deep neural networks. In: Proceedings of the 2018 IEEE International Conference on Service Operations and Logistics, and Informatics (SOLI); 2018. doi: 10.1109/ICEBE.2018.00019.
[20] Khan A, Nazir M, Basit A. Object classification and recognition using Bag-of-Words model. In: 2015 IEEE International Conference on Image Processing (ICIP); 2015. doi: 10.1109/CSPA.2016.7515834.
[21] Richter T, Wolf T, Bocklisch A, Riedl M. Comparing frameworks for building conversational agents: Rasa, Microsoft Bot Framework, Dialogflow, and Wit.ai. In: Proceedings of the 11th International Conference on Natural Language Generation (INLG); 2018. https://www.aclweb.org/anthology/K18-2005.
[22] Hosseini R, Hosseini R, Trudel A. A review of Chatbots in education: Practical examples and research directions. In: Proceedings of the 2019 International Conference on Artificial Intelligence in Education (AIED); 2019. https://link.springer.com/chapter/10.1007/978-3-030-23207-8_1.
[23] Young T, Hazarika D, Poria S, Cambria E. Recent trends in deep learning based natural language processing. IEEE Comput Intel Magazine. 2018;13(3):55–75. doi: 10.1109/MCI.2018.2840738.
[24] Ng SP, Murnane T, Reed K, Grant D, Chen TY. A preliminary survey on software testing practices in Australia. 2004 Australian Software Engineering Conference Proceedings, 2004. doi: 10.1109/ASWEC.2004.1290464.
[25] Subramanian M, Agarwal A, Bajaj P. AI-powered healthcare Chatbots: Transforming patient interaction. npj Digital Med. 2023;6(1):Article 79. https://www.nature.com/articles/s41746-023-00979-5.
[26] Zhou L, Gao J, Li D, Shum HY. The design and implementation of XiaoIce, an empathetic social Chatbot. Comput Linguistics. 2020;46(1):53–93. doi: 10.1162/coli_a_00368.