

Chapter One

Introduction

1.1 Background to the Study

Customer service is critical to the user experience, even more so on the online scene, where most users are extremely demanding in terms of response time and the quality of the responses provided¹. Customer support assists users with product issues and is sometimes a required step when validating an account¹. There are already tools for organising and structuring customer support, such as Zendesk, Freshdesk, and Happyfox¹. In general, these tools provide excellent functionality for defining support levels, evaluating agent performance, and managing communication channels, among other things. However, there is one area where these tools are woefully inadequate: automation². Indeed, while users may submit numerous tickets with varying problem types, writing styles, and vocabulary, most of these tickets refer to a small subset of problems that can be resolved in a systematic manner¹.

As with the well-known Pareto principle, only a small subset of users' problems account for the majority of workload³. Regrettably, the majority of help desk software does not provide satisfactory fully automated solutions to this¹. Partly because it is highly dependent on the type of users who contact the business and because these solutions are rarely custom-built for the target audience in question. Additionally, they are frequently proprietary and difficult to customise. As a result, a customised automation solution is a more appropriate approach. A dialogue system is an example of an intelligent agent that is designed to engage in natural human conversation⁴. More precisely, it is a software application that, with natural language processing and machine learning, stimulates natural language human conversation via text or text-to-speech⁵. Since their inception in the 1960s, dialogue systems have advanced significantly. Computer

science has advanced tremendously in terms of hardware and software, as well as natural language processing and machine learning techniques. Chatbots have evolved from systems that generate machine-like responses to human-like agents capable of developing long-term relationships with users because of the advancement of these emerging technologies. ELIZA and PARRY are two of the most well-known early chatbot implementations⁶. Apple's Siri, Amazon's Alexa, and Microsoft's XiaoIce are all examples of modern chatbots⁷. Chatbots offer a slew of potential advantages. Notably, they provide instant, consistent, and dependable responses and solutions⁸. These characteristics make chatbots an extremely useful tool in a variety of fields, and their use has accelerated in areas such as business, e-commerce, and healthcare⁹.

Chatbots are more formally referred to as conversational agents in the scientific literature. The terms chatbot/conversational agent will be used interchangeably throughout this document¹⁰. Every chatbot's underlying principle is to interact with a human user (in most cases) via text messages and behave as if it could comprehend the conversation and responding appropriately to the user¹¹. In 1950, a simple test dubbed the Turing test was developed in which a human judge was required to predict whether the entity communicating via text was a computer programme or not¹². However, this test's ambition is much greater than the typical use case for chatbots; the primary difference is that a chatbot's domain knowledge is limited, whereas the Turing test assumes that the agent can converse about any subject¹². This aids in the design of conversational agents because they are not required to possess an infinite amount of domain knowledge and can thus focus on very specific topics such as assisting users in booking a table at a restaurant¹⁰.

Additionally, another general assumption that chatbot designers make is that when users initiate an interaction with a chatbot, they typically have a goal in mind for the conversation's

conclusion¹⁰. This then influences the conversation's flow and topics to accomplish the desired outcome. This can be exploited by developers because certain behavioural patterns tend to emerge as a result. Thus, a chatbot can be defined as a computer programme that communicates with human users via text and provides services to them to accomplish a predefined goal¹⁰.

Alan Turing inspired the concept of a chatbot⁶. Turing proposed the Turing Test in 1950, posing the question of whether machines can think. Since then, conversational systems have garnered considerable attention and evolved into a central concept in the field of artificial intelligence¹². ELIZA was the first publicly known chatbot. ELIZA was created in 1966 at the Massachusetts Institute of Technology's Artificial Intelligent Laboratory by Joseph Weizenbaum¹³. It simulates conversations using hand-crafted scripts designed to resemble those of a Rogerian psychotherapist. When a user communicates with ELIZA, he or she types some natural language statements⁶. ELIZA analyses the input text for the presence of a keyword and then generates responses based on a keyword-related rule. While it is capable of conversing, its communication abilities and scope of knowledge are limited^{6,14}.

Several other early chatbots are well-known, including PARRY (developed by Colby in 1975) and ALICE. (created in 2009 by Wallace)^{6,15,16}. Like ELIZA, they use simple pattern-matching algorithms to simulate how a human would behave in a text-based conversation¹⁷. PARRY is regarded as an improvement over ELIZA because it possesses a personality and a more robust control structure⁹. ALICE's development marked another milestone in the history of chatbots¹⁸. It was the world's first online chatbot and won an award for being the most humanoid system¹⁸. ALICE generates responses by matching patterns against 41,000 templates, whereas ELIZA contains only 200 keywords and rules^{6,9,19}. SmarterChild's development in 2001 marked a

significant advancement in chatbot technology. SmarterChild was accessible via messenger applications and could assist users with simple tasks by retrieving data from databases^{9,20}. In 2003, the Defense Advanced Research Projects Agency (DARPA) funded and coordinated the CALO (Cognitive Assistant that Learns and Organizes) project^{6,21}. It was a five-year endeavour to develop a cognitive assistant capable of learning from its users' experiences and performing routine tasks for them. CALO was a seminal project in the history of chatbots because it integrated numerous areas of artificial intelligence and assisted software systems in better understanding human intentions. Additionally, it spawned numerous spin-offs, the most notable being Apple Siri⁶.

IBM's Watson AI project, which has been in development since 2006, made the next advancement for conversational agents²². The agent was created with the sole purpose of winning the American television game show Jeopardy! which it accomplished in 2011 when it faced off against two former champions. Jeopardy! is interesting from an NLP perspective because the questions frequently involve wordplay and necessitate rapid information retrieval from vast knowledge bases. Unfortunately, in its previous incarnation, this AI could only respond to one-liner questions and was incapable of carrying on a proper conversation with another²².

Apple Siri was the first virtual personal assistant, launched in 2010. Other assistants such as IBM Watson, Microsoft Cortana, Amazon Alexa, and Google Assistant have since followed⁶. The virtual personal assistants were integrated into smartphones or smart speakers and were capable of understanding and responding to human speech, as well as performing more advanced tasks. In comparison to earlier systems, virtual personal assistants are connected to the internet and thus

capable of responding very quickly⁶. However, miscommunications occur frequently because they do not understand colloquial languages and cannot interpret the input within the context of the dialogue⁶. Another notable chatbot is Microsoft's XiaoIce. It is a social chatbot, demonstrating that conversational agents are capable of not only conducting conversations and performing simple tasks, but also satisfying our need for sociability^{6,23}. Recent advancements in chatbot technology include the development of open-domain chatbots⁶. Google's Meena and Facebook's Blender are both dialogue systems capable of conversing about virtually anything and performing at or near human levels in a variety of areas⁶.

Chatbots can be classified according to a variety of factors, including the service provided, the domain of knowledge, the method used to generate responses, the goal, and the permission, as well as the amount of human assistance^{6,9}. Classification according to the service provided considers the chatbot's sentimental proximity to the user, the amount of intimate interaction that occurs, and the task performed by the chatbot. Interpersonal chatbots provide services such as booking and FAQ searches without acting as the user's companion. Intrapersonal chatbots operate within the user's personal domain and are expected to understand the user in the same way that humans do^{9,24}. Inter-agent chatbots interact with other chatbots to complete a task⁹. The knowledge domain denotes the body of knowledge that chatbots have access to or the volume of data on which they are trained. While open domain chatbots can respond to questions from any domain, closed domain chatbots are limited to a single domain^{9,25}. Classification based on response generation method takes response generation into account²⁶. A chatbot can be rule-based, retrieval-based, or generative-based²⁷. Classification according to goals considers the primary objective of a chatbot. Chatbots that provide specific information to the user are stored in a fixed source⁶. Chat-based or conversational chatbots are designed to converse with the user

in the same way that another human would^{9,28}. Chatbots that are task-based carry out a task in response to the user's requests⁹. Additionally, chatbots can be classified based on the permissions granted by the development platforms⁹. Open-source platforms make their source code available to developers, giving them complete control over the majority of implementation details⁹. Closed platforms are composed entirely of proprietary code and are typically offered by large businesses⁹. Chatbots can be classified as human-mediated or fully autonomous, depending on the amount of human assistance. At least one component of a human-mediated chatbot is composed of human computation^{6,29}. While fully automated chatbots process information quickly, human computation adds flexibility and can help fill in gaps caused by algorithm limitations⁶.

1.2 Statement of the Problem

Due to the increased user population of web-based financial and payment application services, users are increasingly confronted with long queue, service malfunctions such as traffic, server errors, payment errors, and system override. Customers preferred to solve transaction issues, make enquiries, and forward complaints remotely in a preferred language. Many chatbots have been developed for different purposes including financial transactions^{1,5,14}. However, to the best of the researcher's knowledge, there is a handful of development of chatbots that translate from any language to English for the bot to understand and send the response in the preferred language of the user has not been explored in the extant literature, especially in the financial sector.

In this thesis, the author tends to advance this position by using Artificial Intelligence (AI) in an intelligent chatbot setting to provide support, personalised assistance, and remote enquiring possibility in a preferred language using frequently asked questions (FAQs) thereby automating

certain aspects of customer support and offload some of the workload from support service agents.

1.3 Aim and Objectives of the Study

The aim of this study is to design a chatbot to enhance web-based financial and payment application services.

Specific Objectives are to:

- i. develop a web page on which the chatbot was deployed using Java interpreter and spring boot project using Spring initializr
- ii. implement the design in (i), the Bot logic and build a web UI to interact with the AIML bot using vaadin
- iii. Evaluate and test the performance of the designed chatbot using accuracy, scalability, and functionality.

1.4 Significance of the Study

This study's objective is to develop an automated customer support system. This study will contribute significantly to the existing body of knowledge, by assisting in the development of customer support applications that reduce customer complaints and wait times.

Stakeholders in the digital economy, information technology firms, and private and public organisations can use the study's findings to inform technological decisions that will result in improved customer experiences. Additionally, this study will serve as a resource for students

enrolled in Nigerian universities who are studying computer science, artificial intelligence, and related fields.

Additionally, the study's findings will serve as a reference for computer science students, lecturers, and researchers. It may act as a catalyst for further research on the subject. Additionally, findings may result in the development of novel ideas for the development of chatbots in the field of artificial intelligence.

1.5 Scope of the Study

The scope of this thesis is to conceptualise, implement, and evaluate a conversational agent (CA) for use in an automated customer support system. The chatbot can provide a scalable machine learning pipeline for the system's models, starting with raw data, and can respond to users in real-time, assisting users with a subset of common problems, allowing for tuning of the bot's responses, and supporting multiple languages. The first step is to create a conversational flow for the chatbot's interaction. Prior to implementation, the available chatbot technologies will be evaluated by weighing the benefits and drawbacks of various frameworks. The chatbot will be evaluated based on its usage and ability to enhance the customer experience. The findings will be presented and interpreted descriptively. The researcher's objective is to deduce patterns in the bot's interactions that can be used to inform future development and improvement.

1.6 Limitation to the Study

This study was limited to the design of an intelligent chatbot setting to provide support, personalised assistance, and remote enquiring possibility in a preferred language using frequently asked questions (FAQs).

Also, this study was limited by the tools used. Most are open source, hence, other cost related applications may give a better performance and results. Also, the choice of network for internet connectivity is also a limitation, since the design must give real time responses which will depend on the internet speed and choice network. Training the system's models will also take a reasonable amount of time.

1.7 Operational Definition of Terms

Artificial Intelligence (AI): Computerized technology that is used to collect and process data and is capable of learning and applying constantly evolving logic and rationale in a manner like human intellect.

Artificial Intelligence Marketing (AIM): Utilization of artificial intelligence technology to leverage massive amounts of customer data acquired through interaction with the digital platform, processed via machine learning, and applied to anticipate, predict, and respond to customer behaviour in a manner that mimics human to human interaction.

Chatbots: Automated programmes that make use of artificial intelligence technology to interact with human users of digital platforms such as web-based input forms and instant messengers. The chatbot employs machine learning to develop the ability to predict users' behaviour and respond to input stimuli in a manner like human-to-human conversation.

Customer Experience: Considered to encompass all aspects of a customer's interaction with a business and includes user experience as a constituent.

User Experience: Every aspect of a product or service's end user. UX is considered all-encompassing, encompassing concepts such as ease of use, ergonomics, functionality, emotional response, and engagement, as well as the way the end user experiences and perceives the interaction.

Customer Engagement: Active relationship between brands and their customers, which includes interactions that raise awareness, foster loyalty, and foster long-term relationships.

Do Not Copy, Lead City University, Nigeria

Endnotes

- ¹ F Peters. Master thesis: *Design and implementation of a chatbot in the context of customer support*. Masters Dissertation.2018.
- ² A Truong, A Walters, J Goodsitt, K Hines, CB Bruss, & R Farivar. *Towards automated machine learning: Evaluation and comparison of AutoML approaches and tools*. In2019 IEEE 31st international conference on tools with artificial intelligence (ICTAI) 2019 Nov 4 (pp. 1471-1479). IEEE.
- ³ J Barber, X Yu, LK Zamore, J Lin, V Jazayeri, S Erlich, T Savor, & M Stumm. *Bladerunner: Stream processing at scale for a live view of backend data mutations at the edge*. InProceedings of the ACM SIGOPS 28th Symposium on Operating Systems Principles 2021 Oct 26 (pp. 708-723).
- ⁴ E Merdivan, D Singh, S Hanke, & A Holzinger. *Dialogue systems for intelligent human computer interactions*. Electronic Notes in Theoretical Computer Science. 2019 May 4;343:57-71.
- ⁵ T Schachner, R Keller, & F von Wangenheim. *Artificial intelligence-based conversational agents for chronic conditions: systematic literature review*. **Journal of medical Internet research**. 2020 Sep 14;22(9):e20701.
- ⁶ X Huang, A CIS. *Chatbot: Design, architectre, and applications*. Senior Capstone Thesis University Of Pennsylvania | School Of Engineering And Applied Science May 3, 2021
- ⁷ B Olivier. *The 'Transhuman'Otherness of Artificial Intelligence (AI): Gibson's Agency, Jonze's Her and Brown's Origin*. Alternation Special Edition 38: 302 - 344 (2021) 302 Print ISSN 1023-1757; Electronic ISSN: 2519-5476; DOI <https://doi.org/10.29086/2519-5476/2021/sp38a13>
- ⁸ X Cheng, X Zhang, J Cohen, & J Mou. Human vs. AI: *Understanding the impact of anthropomorphism on consumer response to chatbots from the perspective of trust and relationship norms*. Information Processing & Management. 2022 May 1;59(3):102940.
- ⁹ E Adamopoulou, & L Moussiades. *An overview of chatbot technology*. InIFIP International Conference on Artificial Intelligence Applications and Innovations 2020 Jun 5 (pp. 373-383). Springer, Cham.
- ¹⁰ F Peters. Master thesis: *Design and implementation of a chatbot in the context of customer support*. Masters Dissertation.2018.

11. A Rapp, L Curti, & A Boldi. *The human side of human-chatbot interaction: A systematic literature review of ten years of research on text-based chatbots*. **International Journal of Human-Computer Studies**. 2021 Jul 1;151:102630.
12. S Natale. *Deceitful media: Artificial intelligence and social life after the Turing test*. Oxford University Press, USA; 2021 Mar 17.
13. C. OT, H EL, P OU, U ER, KE I'DLI, & T ER. *The Evolution Of Customer Service Chatbots*. NCT, November, 202 1
14. R Winkler, & M Söllner. *Unleashing the potential of chatbots in education: A state-of-the-art analysis*. In Academy of Management Annual Meeting (AOM) 2018.
15. S Dumpala. *Check-It-Chatbot* (Doctoral dissertation, Dublin Business School).2021
16. NP Naik. *Performance Measurement of Natural Dialog System by Analyzing the Conversation*. In Designing User Interfaces With a Data Science Approach 2022 (pp. 180-209). IGI Global.
17. A Soufyane, BA Abdelhakim, & AB Ahmed. *An intelligent chatbot using NLP and TF-IDF algorithm for text understanding applied to the medical field*. In *Emerging Trends in ICT for Sustainable Development 2021* (pp. 3-10). Springer, Cham.
18. S Cunningham-Nelson, W Boles, L Trouton, & E Margerison. *A review of chatbots in education: practical steps forward*. In *30th Annual Conference for the Australasian Association for Engineering Education (AAEE 2019): Educators Becoming Agents of Change: Innovate, Integrate, Motivate 2019* (pp. 299-306). Engineers Australia.
19. J Lannoy. *The effect of chatbot personality on emotional connection and customer satisfaction* (Master's thesis, University of Twente) 2017.
20. S Paliwal, V Bharti, & AK Mishra. *Ai chatbots: Transforming the digital world*. In *Recent Trends and Advances in Artificial Intelligence and Internet of Things 2020* (pp. 455-482). Springer, Cham.
21. W Barfield, & U Pagallo, editors. *Research handbook on the law of artificial intelligence*. Edward Elgar Publishing; 2018 Dec 28.
22. M McTear. *Conversational AI: dialogue systems, conversational agents, and chatbots*. *Synthesis Lectures on Human Language Technologies*. 2020 Oct 30;13(3):1-251.
23. L Zhou, J Gao, D Li, & HY Shum. *The design and implementation of xiaoice, an empathetic social chatbot*. *Computational Linguistics*. 2020 Mar 1;46(1):53-93.

- ²⁴. J Fernández-Avelino, G Alor-Hernández, MA Paredes-Valverde, & LN Sánchez-Morales. *Developing Chatbots for Supporting Health Self-Management*. In *Handbook of Research on Natural Language Processing and Smart Service Systems 2021* (pp. 135-156). IGI Global.
- ²⁵. J.A Kapočiūtė-Dzikienė. *domain-specific generative chatbot trained from little data*. *Applied Sciences*. 2020 Jan;10(7):2221.
- ²⁶. HY Shum, XD He, & D Li. *From Eliza to Xiaolce: challenges and opportunities with social chatbots*. *Frontiers of Information Technology & Electronic Engineering*. 2018 Jan;19(1):10-26.
- ²⁷. S Hussain, O Ameri Sianaki, & N Ababneh. *A survey on conversational agents/chatbots classification and design techniques*. In *Workshops of the International Conference on Advanced Information Networking and Applications 2019 Mar 27* (pp. 946-956). Springer, Cham.
- ²⁸. RM Schuetzler, GM Grimes, JS Giboney, & HK Rosser. *Deciding Whether and How to Deploy Chatbots*. *MIS Quarterly Executive*. 2021 Mar 1;20(1).
- ²⁹. Q Motger, X Franch, & J Marco. *Conversational Agents in Software Engineering: Survey, Taxonomy and Challenges*. arXiv preprint arXiv:2106.10901. 2021 Jun 21.

Do Not Copy, Lead City University, Nigeria

Chapter Two

Literature Review

2.1 Conceptual Review

Chatbots give the impression of conversing with humans while communicating with computer software that is brought to life by natural language input¹. Others simply define it as a computer software that uses artificial intelligence to mimic user talks. A chatbot is a piece of software that allows for text-based communication utilising natural language². It appears that simulating actual persons is vital for user acceptance of chatbots, which underscores the critical relevance of a big knowledge base, i.e., the existing set of rules a chatbot has³. With more and more advertising reaching and overwhelming customers both online and in person, contacting them becomes an increasingly tough endeavour. Chatbots may soon become one of the best ways for businesses to communicate with individual customers and will determine a company's competitiveness⁴. Furthermore, increasing interest in chatbots has significantly ascribed to key changes involving the rise of messaging services and breakthroughs in Artificial Intelligence⁵.

2.1.1 Human-Computer Interaction

Human-computer interaction necessitates the precise development of information systems that assist users in performing tasks in a more productive and correct manner⁶. This is important for organisations since it is critical that newly introduced technology is widely accepted by users in real life. New technological changes will only be adopted if they are adequately planned. This does not necessarily apply to every single person, but it is critical to ensure that it welcomes the individuals for whom it was designed, emphasising the need of usability testing⁶. The two key

principles to ensuring effective human-computer interaction are visibility and affordance, which means that technology should be apparent and functional to a user. As previously stated, this stresses the importance of usability⁷. The ease with which technologies can be learned and used is referred to as usability⁸. If a computer system is badly built, users will encounter problems and eventually abandon it. However, strong usability does not always indicate that a computer system has a wide range of functionality.

A firm aims to provide good usability by understanding people's characteristics and how they use technology, developing tools to design computer systems that are suitable for the tasks they are meant for, and achieving an efficient and successful human-technology interaction⁹. Taking all of these factors into account, the user should always come first. Metrics can be used to ensure that a computer system operates in accordance with previously established criteria. Metrics are just numerical measures⁷. There are many various types of metrics that can be utilised, depending on their intended function. The four primary classes are as follows⁷:

- i. Duration metrics are used to determine how much time is spent on a particular activity.
- ii. Count measures ensure that the number of times a specific action occurs is recorded.
- iii. The proportion of tasks done is not a straightforward assessment; however, the basic idea is to first create goals and then quantify how frequently the intended outcome was attained.
- iv. Similarly, because output is made up of many components, measuring its quality is difficult. One could argue that distinguishing between good and terrible quality is simple; yet, when it comes to human-computer interaction, the user's impression is crucial.

2.1.2 Components of Chatbot

Chatbots, in general, can be thought of as a combination of three parts: the interface, which is the interface between the chatbot and human users via a messaging app or a chat session on a company's website, the intelligence, which allows the chatbot to understand and solve customer queries and learn from each interaction, and the integration, which primarily covers integration with other systems and platforms^{9,10}. In addition, chatbots have inputs and outputs. Inputs are best described as recording and entering data into a system and delivering instructions to the system¹¹. In order for a chatbot to function, the user's input into the conversation must match one of the pre-defined inputs so that the computer system can correctly interpret it¹². As a result, the three important factors are as follows: A chatbot must be able to adapt to the characteristics of its user, be appropriate for the activities it is intended to perform and be appropriate for the job and environment for which it is developed. Following implementation, ensure that relevant input is obtained to be able to analyse the system and adapt appropriate changes⁷.

Telephone calls and face-to-face interactions are no longer the only ways to communicate with clients. Chatbots' benefits are becoming more apparent as they gain popularity¹³. The fundamental benefit of chatbots is their constant availability. Chatbots are always available to answer users' questions. Customers expect accurate, personalised, and timely responses when contacting a company, and chatbots will meet those expectations because to their broad knowledge and availability¹⁴. Furthermore, the issue of needing to wait until one gets through to an agent is resolved because most chatbots allow for the handling of infinite queries at the same time without generating issues. However, it is preferable to inform consumers that they are conversing with a computer rather than trying to conceal and fool them¹³. When buying online,

chatbots can improve cross- and upselling and assist buyers in discovering the appropriate item¹³. Chatbots can then provide recommendations based on previous information, preferences, and order histories. This will also enable one-click ordering the next time, boosting user convenience⁷.

Furthermore, there would be no need for answering FAQs because chatbots will serve as a replacement for them, automating the process while also giving clients with relevant replies and enhancing customer satisfaction¹³. Deploying bots on e-commerce platforms is advantageous since it can also handle returns, decreasing staff costs while increasing consumer happiness and loyalty¹⁵. The expected annual savings owing to the usage of AI bots will be more than \$8 billion by 2022^{7,16}. Nowadays, it is simple to identify an organization's target demographic and promote effectively to them. Chatbots can operate as a subscription tool and, for example, provide fitness suggestions, allowing a business to promote and sell an appropriate item or product while leveraging this subscription chatbot for affiliate marketing^{7,17}.

The four key benefits of chatbots are increased convenience, cost savings, the ability to maximise client interaction and outreach, and the reduction of customer care man hours¹⁵. Chatbots could aid the travel industry, since they can serve as 24-hour customer service representatives and, if properly set up, can function without the need for human interaction. Customers will also be served more rapidly because the time spent waiting for an enquiry to be answered is minimised, and chatbots deliver instant answers. Most humans are amenable to using chatbots, which could imply that if used consistently by customers, a company will be able to profile clients and collect rich qualitative data, allowing the detection of trends among the

firm's target group⁷. Additionally, following the talk, a business could send out feedback forms or surveys to get additional information on how to improve the guest's experience.

Chatbots, like any other kind of technology, have several drawbacks and possible risks⁷. Heterogeneity issues have a significant impact on any service. Customers are individuals with distinct identities and needs, allowing service delivery to be highly personalised and adapted to the exact individual. This is also represented by the need for individualization. Furthermore, it is critical not only to focus on service quality consistency to avoid falling into a vicious circle in which poor service quality might potentially kill a firm, but also to provide service recovery for any unsuccessful services⁷. All these factors apply to all internet services, not only chatbots. Almost all service procedures must cope with the problem of simultaneity or inseparability⁷. This encourages the fact that most services are generated and consumed at the same time, implying that production will not begin prior to the user's input provision⁷.

Because chatbots are designed by developers, they are pre-coded and will only recognise an input if it corresponds to an expected path¹⁸. When something that does not follow the expected path is entered into the chatbot, the output is dramatically influenced¹⁸. As a result, the output will be either repeated or unpleasant to the consumer, failing to provide the correct response and leaving people dissatisfied¹⁵. One big issue with aligning input provision forms sarcasm and irony is that chatbots appear to be unable to interpret sarcasm correctly, mistaking it for genuine¹⁹. The comfort of purchasing online and locating things for a consumer via chat²⁰ is one advantage of chatbots. However, this could cause confusion and influence the customer's purchasing decisions. Having to narrow down each filter parameter, such as size, colour, kind, and material, could become a time-consuming operation, limiting client convenience, and

removing the benefit of this aspect⁷. Because of the reduction in customer service man hours, chatbots may contribute to a decrease in labour demand¹⁵. The most affected positions will be those with low-level and repetitive duties, which will be replaced by a chatbot as a standardised process. In the long run, the threat of declining demand for low-wage occupations will be especially severe in developing nations, where a high proportion of the population works in such a job⁷.

Since the global use of chatbots has just recently increased, the threat of social engineering attacks has been very high²¹. The example of Microsoft Tay, which began tweeting racist tweets mere days after its inception, indicates that hackers attempted to exploit chatbots to propagate racist, sexist, or other harmful news²². Furthermore, some users' willingness to utilise a chatbot may be hampered because chatbot data processing appears to be a problem that has yet to be resolved⁷. Users must be able to trust the chatbot, and any data, particularly personal data, must be treated securely⁷. Although the travel sector could benefit from adding chatbots, according to the Kayak Mobile Travel Report, many British citizens are still sceptical about using chatbots⁷. 34 percent are concerned about data security, and nearly a third are apprehensive that the chatbot will not understand the query sent¹⁵. Furthermore, 33% believe that a chatbot would offer them an incorrect answer²³. These factors may limit the extent to which chatbots are adopted and employed by clients.

Measuring the economic worth of a chatbot is complex and varies according on industry and organization⁷. One method would be to measure the rise in sales for a store or the change in website visitors who booked a night at a hotel, for example. The acquired data should be compared to the previous year in which the chatbot was not yet activated. The data collection

method is time-consuming and resource-intensive¹⁵. What may be discovered to be probable to enhance is that an industry's chatbot installation may result in a rise in revenue. A chatbot, on the other hand, can aid with customer service by providing information and assisting owners in detecting a client's viewpoint and expectations⁷.

Furthermore, business owners will be able to identify any holes in their website. Even if a conversation does not result in an immediate purchase, a chatbot will keep useful information about potential clients and is thus economically valuable¹⁵. When humans communicate with a chatbot rather than another human, the average words per message length is shorter, but the average messages sent each discussion is higher^{7,24}. Furthermore, there is a difference in the use of profanity and foul language: both variables were higher during the human-chatbot conversation compared to the human-human conversation, indicating that humans communicate differently with chatbots than with real humans and are more uncomfortable talking to a chatbot. This could have a detrimental impact on a customer's propensity to utilise a chatbot⁷.

2.1.3 Capabilities and Metrics

Some chatbot capabilities, functionalities, and metrics are utilised in this study to assess the effectiveness and utility of several chatbot platforms²⁵. To begin, some metrics that might be used to measure a chatbot's usefulness are the number of matches per question, which is the number of evaluators who were able to get responses from the chatbot per question asked, and the number of matches found per user or in total²⁵. Furthermore, efficient chatbots allow you to schedule postings and messages. They will assist in the planning of a company's marketing activities and will inform users of changes or news such as special deals, promotions, discounts, or new inventory. A chatbot should offer the firm with summaries of the talks and statistical data

in order for the company to be able to measure the results of the chatbots and whether the discussions were effective. Chatbots can be used by businesses to perform data comparison and extra data analytics²⁶.

2.1.4 Use of Chatbot

2.1.4.1 Customer Service

Chatbots are a new type of customer care that is transforming the way a client's after-sales experience is shaped²⁷. Chatbots are important for customer service for three reasons²⁸. To begin, while a chatbot cannot address all queries, routine queries, which account for most service requests, can be standardised and delivered at a high degree of performance^{7,28}. To achieve this high-quality output, the chatbot must be trained for approximately five weeks⁷. Furthermore, chatbots make it simple and quick for clients to contact customer support through messaging systems that they already use on a daily basis⁷. As a result, many users will be comfortable with this form of interaction, making it a convenient experience. Finally, chatbots will give businesses insight into their digital customers and keep them up to date on new technology⁷.

Aside from the general use of chatbots, one virtual agent was deployed especially for tourism purposes: CLARA²⁹. CLARA was designed to function as both a conference information system and a local tour guide, both of which would undoubtedly provide important information and assistance to first-time visitors⁷. The system architecture is made up of three components: the client system, a web socket server, and various information resources. A user enters a query using the application's graphical interface, and the system generates a JSON message that includes the query, the domain, which determines whether the information is conference or touristic, and GPS coordinates, if the user enables the system to retrieve them⁷. First, the algorithm searches the index for generic queries like Elizabeth's and retrieves the most similar

examples³⁰. If no answer is in the index, the system will begin looking for answers on the conference search engine or the tourist search engine. For all types of inputs, this cycle is quite similar. To deliver a final answer, the system generates a JSON message comprising information about the feedback, the kind, as an answer could be a map, an external website, or anything else, and the URL to be displayed⁷. When retrieving conference information, the system allows the user to search for authors, affiliations, countries, titles, conference sessions and events, and general conference facility questions⁷. The input will be recognised by the system using fuzzy search methods that are resistant to misspellings⁷. Following that, the entities take knowledge from the knowledge bases, and the SQL query is constructed and utilised to search the database for publications, authors, or session information⁷. CLARA was launched during the 15th International Speech Communication Association conference in 2014, with 1200 delegates from 46 different countries³¹. To accomplish this, a free mobile application was developed for both the Google Play Store and the Apple App Store that could display information about authors, the conference schedule, a conference venue map, provide access to the conference's website, and even allow users to add the conference schedule to their personal calendar³¹. The application also provided access to the agent screen. It was discovered that, in general, people would prefer to use the agent screen to acquire tourist information rather than conference information³¹.

However, it was discovered that the system could not answer a high percentage of queries, approximately 50%, but the majority of these, 75%, were concerning out-of-domain queries during chat interactions, for which the system had problems detecting an answer from the index or external websites and was thus unable to process them correctly³¹.

2.1.4.2 Tourism and Hospitality

With the advent of digitalisation, an increasing number of sectors are shifting their primary attention to online applications and the internet. The hotel sector estimates that 75% of all bookings are made online, resulting in an increase in demand for online information provision³². Incorporating a hotel chatbot can determine a hotel's competitiveness on the market^{7,32}. A hotel chatbot is designed primarily to achieve a certain marketing purpose, such as increasing revenue. As a result, its knowledge base is restricted to hotel information and touristic issues, providing users with information on the hotel itself, the hotel's surrounds, and answers to the most common types of touristic queries⁷. The introduction of a chatbot on a hotel's website yielded some significant results. The average chatbot conversation time was 4.2 seconds, with a maximum of 118 seconds, whereas 36% of all interactions had only one user input and 60% had no more than two user inputs⁷. In compared to more general chatbots, hotel chatbot talks are relatively short, which could be explained by the hotel chatbot's limited knowledge base and the lack of small talk. Furthermore, another view is that a hotel chatbot is only an adjunct to a hotel's website, not a standalone tool for reserving a room^{7,33}. According to the findings, 56 percent of all discussions involved questions about the hotel or its offerings, while 12 percent involved questions about the chatbot itself⁷.

2.1.5 History of Chatbot

2.1.5.1 ELIZA

The ELIZA effect was supported by most users who saw the hotel chatbot as a person⁷. Expedia, according to reported, has been a model of consumer involvement through the employment of a chatbot⁷. Customers can book a hotel through its Facebook Messenger chatbot and are merely sent to the website to submit payment information^{34,35}. Another success story of chatbot

application is Kayak, a travel search site that provides information on flights, hotels, rental vehicles, and package holidays⁷. Kayak's chatbot not only provides complete information on these categories, but it also recommends other destinations based on the user's preferences and provides future updates to the client via Facebook Messenger^{7,34}. Other organisations adopting chatbots in tourism and hospitality include the two airlines KLM and Icelandair, Bebot, which provides chatbot concierge for Japanese travellers, and Aloft Hotels, which uses a chatbot to provide guests with an extra option to make service requests⁷.

The purpose of the very first chatbot produced was solely to entertain and engage users. In the 1960s, a chatbot named ELIZA was created to serve as a substitute for a psychotherapist in clinical treatment^{7,36}. Due to its employment as a psychotherapist, users frequently have a very emotional attachment with the computer, as characterized by the ELIZA-effect, which attributes human intelligence to chatbots⁷. ELIZA employs keyword matching, which implies that following a particular input, the programme will check for keywords that match. If matching keywords are located, the system will generate an answer based on the rules specified for this term; otherwise, a connected remark is recalled³⁶. Consequently, ELIZA does not necessarily comprehend the user's difficulties; she just matches their responses to her usual responses⁷.

2.1.5.2 Elizabeth

Elizabeth is an adaptation of ELIZA that has been upgraded to provide greater flexibility and, hence, a greater likelihood of producing an accurate and relevant response⁷. Elizabeth saves information in a script file format, with each line beginning with a script command notation to differentiate between the many categories of knowledge⁷. The following are the notations: W

represents a welcome message; Q represents a quit message; V represents a void input; I represents an input transformation; and K represents a key word pattern.

2.1.5.3 ALICE

ALICE, the Artificial Linguistic Internet Computer Entity, was constructed and utilised for the first time in 1995³⁷. ALICE retains its knowledge of English conversation patterns in AIML files, Artificial Intelligence Markup Language, as opposed to ELIZA's basic keyword matching³⁸. These AIML files consist of data items known as AIML objects, which are composed of themes and categories⁷. The subjects have a name attribute and a set of associated categories, whereas categories are the fundamental unit of knowledge in AIML. Each category acts as a rule for aligning the user's input with the desired result, as well as a pattern and template⁷.

2.2 Methodological Review

Conceptually, a chatbot consists of numerous components collaborating to achieve a single purpose. Figure 2.1 is a graphic summary of the relationships between each component of a conversational agent³⁹. When a new message is received, it is initially handled by the language identification module. This can range from a straightforward tag retrieval to more complex statistical techniques. The new message, together with the language and potential past discussion messages acquired from the backend, are then passed to the intent classifier module, whose function is to infer the intent the user is attempting to communicate³⁹. The message's metadata, inferred intent, and other information from the backend will then be used to select an appropriate action or series of actions. For instance, if the intent is still unclear, a chatbot may elect to respond with a question, or it may decide to reactivate a user account if the user's intention is to request permission to revive the account³⁹.

Finally, the action handler module accepts an action as input and executes said action correctly. It is advantageous to do it in this manner, as the same operation can be executed in various ways based on the agent's environment³⁹. Depending on whether a bot works on the Messenger platform or a company's website, the manner in which an action is carried out may vary drastically.

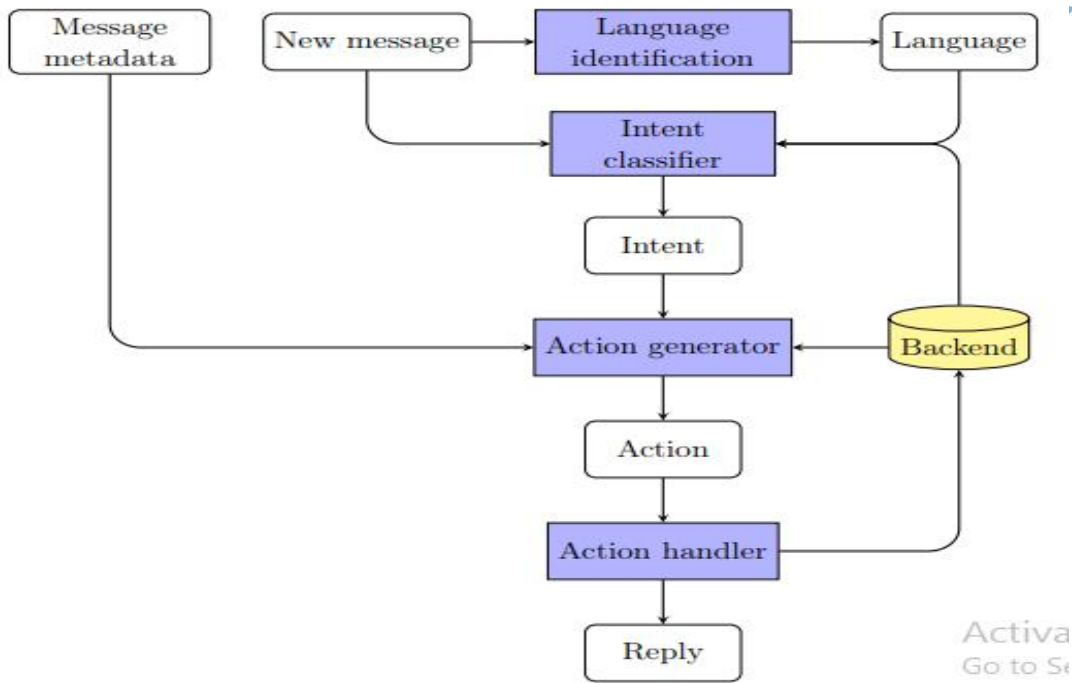


Figure 2.1: High-level Diagram of a Chatbot¹.

2.2.1 Common Deep Learning Techniques

Word Embedding

In essence, word embedding is the process of mapping words to vectors. This vector representation can then be directly provided as features to an algorithm for machine learning⁴⁰.

There are numerous ways to do this function, ranging from a straightforward count vector to deep learning techniques such as Word2vec and GloVe³⁹. Skip-gram model the skip-gram

model's core premise was first proposed in Efficient Estimation of Word Representations in Vector Space: train a shallow neural network with a single hidden layer of fixed size to predict a context given a word³⁹. The architecture of the skip-gram model is depicted in Figure 2.2. Essentially, artificial training examples of the form $(w_t, [w_{t-2}, w_{t-1}, w_{t+1}, w_{t+2}])$ ³⁹. The definition of context changes among writers, but the underlying premise remains the same: a fixed-size sliding window is slid across the sequence of words; the middle of the window corresponds to the target word, and the words before and following it create the context for that word. The ultimate objective is to retrieve the innermost layer and utilise it as the vocabulary's vector representation. Indeed, if the input words are one-hot encoded, it functions solely as a vector representation lookup table. In reality, this entails first converting the corpus into sequences of indices, then training the skip-gram model on those sequences, and finally converting the sequences of indices into sequences of vectors to train the final algorithm³⁹.

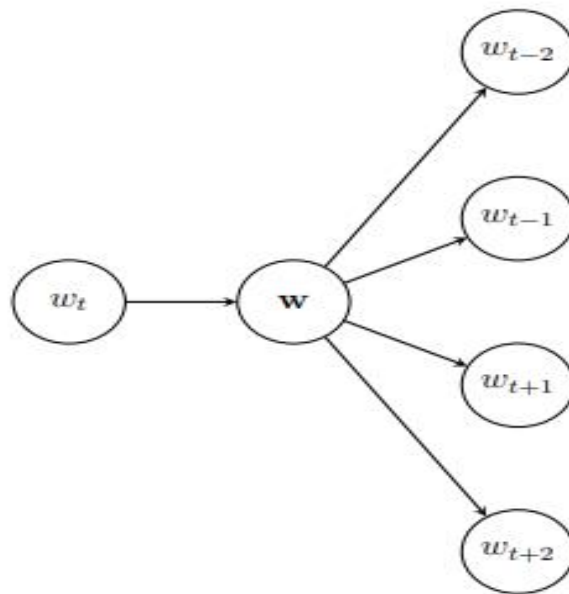


Figure 2.2: Skip-Gram Model in the Case of a 5 Words Window¹.

Recurrent Neural Networks RNNs

Recurrent Neural Networks are a subtype of neural networks designed specifically for data sequences. In essence, they are neural networks containing loops^{39,41}. They have an internal state, commonly labelled C_t for the state at time t , which is transmitted back into the neuron during the subsequent timestep³⁹. At each timestep, the neuron additionally produces a value denoted by h_t . Nevertheless, the primary issue with naive RNN implementations is that they are severely affected by the disappearing and exploding gradient problems³⁹. Several variations of RNNs have been presented as potential solutions to this problem, employing a variety of techniques. In the following paragraphs, the most prevalent ones will be discussed. Units of Long-Short-Term Memory In Natural Language Processing, Long-Short-Term Memory units (LSTMs for short) are often utilised⁴². They were first introduced in 1997 and have gained popularity in recent years because of developments in hardware-accelerated deep learning³⁹. In addition, they have demonstrated encouraging results in machine translation and image captioning. Figure 2.3 illustrates an example of an unrolled LSTM layer. This section explains the forget gate, input gate, cell state, and output gate, which are the primary components of an LSTM unit. Let $x = (x_0, x_1, \dots, x_\tau)^T$ be the input sequence, and $h = (h_0, h_1, \dots, h_\tau)^T$ be the hidden state produced by the LSTM layer, then the relation between x and h is given by the following equations³⁹:

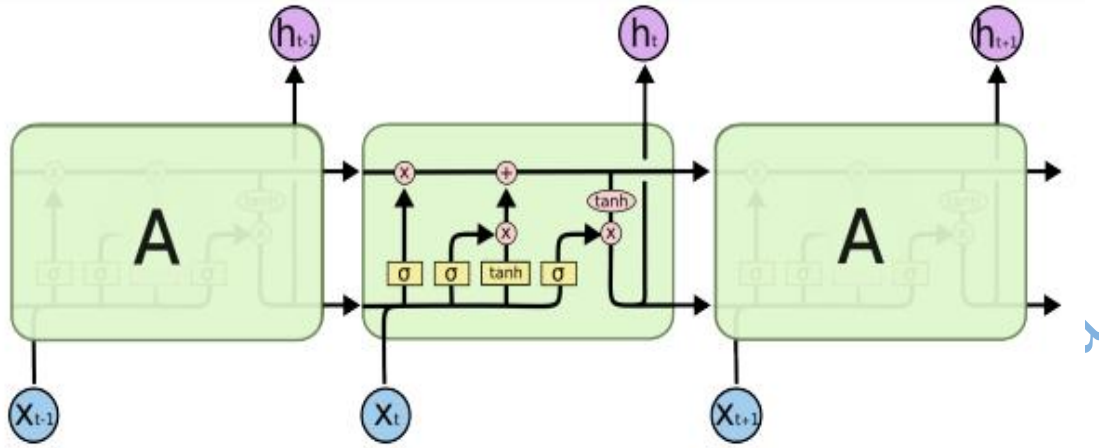


Figure 2.3: Unrolled LSTM layer made of 3 Units⁶.

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \quad (2.1)$$

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \quad (2.2)$$

$$C_t = f_t C_{t-1} + i_t \tanh(W_C \cdot [h_{t-1}, x_t] + b_C) \quad (2.3)$$

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \quad (2.4)$$

$$h_t = o_t \tanh(C_t) \quad (2.5)$$

Equation 2.1 is the forget gate's output. Intuitively, it represents which information the unit will keep or forget from the previous cell state C_{t-1} as it is directly multiplied with it. Figure 2.4 illustrates this process³⁹.

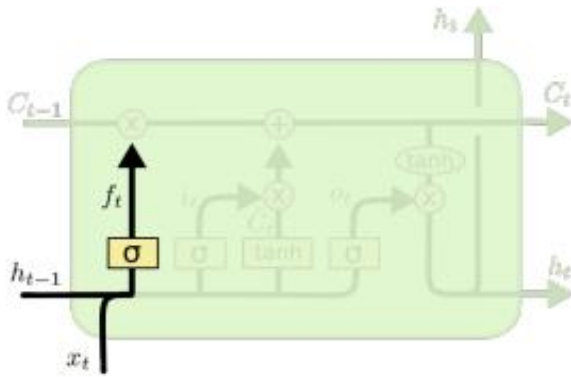


Figure 2.4: LSTM Unit's Forget Gate⁶

The input gate's response is given by equation 2.2. It simply creates a new candidate value for the cell state, denoted C_t , scaled by i_t as shown on Figure 2.5³⁹

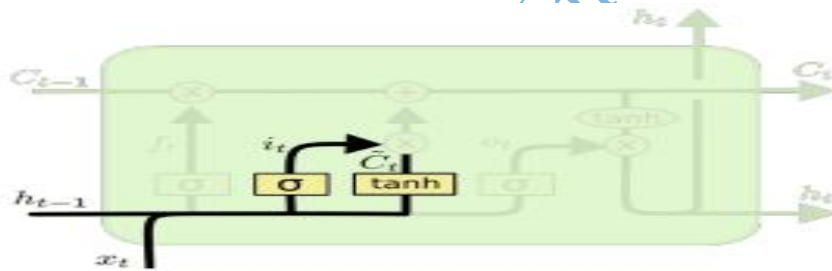


Figure 2.5: LSTM Unit's Input Gate⁶.

The state of the cell is then modified in accordance with equation 2.3. It combines the effects of the input and forget gates. In Figure 2.6, the pertinent relationships for this upgrade are underlined.

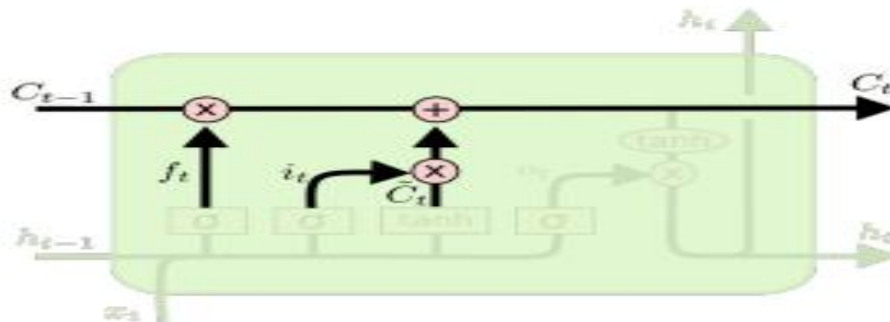


Figure 2.6: LSTM Unit's Cell State Update⁶

Once the cell state is updated, the output gate's activation can be computed using equation 2.4 to generate the unit's hidden state h_t . This is shown in Figure 2.7.

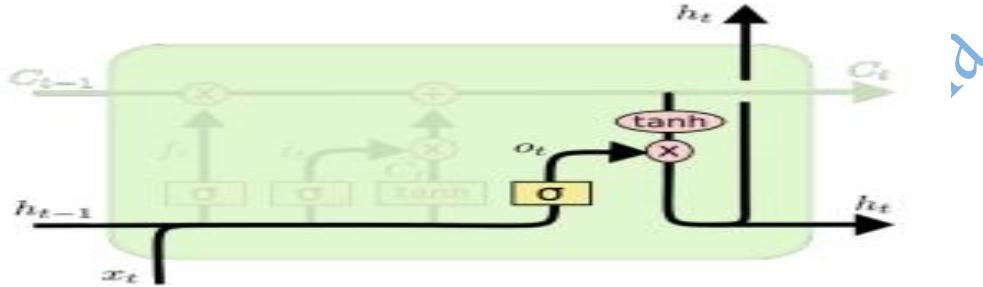


Figure 2.7: LSTM Unit's Output Gate⁶.

Gated Recurrent Units

GRUs are a version of LSTMs in which the forget and input gates are merged into a single update gate, together with the cell and hidden states⁴⁴. In comparison to LSTMs, GRUs have fewer parameters to tune, resulting in shorter training times^{39,44}. However, it has been demonstrated that their performance is comparable to that of traditional LSTM networks³⁹.

Below are provided, for completeness, the equations for GRUs:

$$z_t = \sigma(W_z \cdot [h_{t-1}, x_t] + b_z) \quad (2.6)$$

$$r_t = \sigma(W_r \cdot [h_{t-1}, x_t] + b_r) \quad (2.7)$$

$$\tilde{h}_t = \tanh(W \cdot [r_t h_{t-1}, x_t]) \quad (2.8)$$

$$h_t = (1 - z_t)h_{t-1} + z_t \tilde{h}_t \quad (2.9)$$

where z_t and r_t correspond to the GRU's inner gates as shown in the diagram in Figure 2.8. Both the LSTM and GRU were considered for developing the chatbot³⁹.

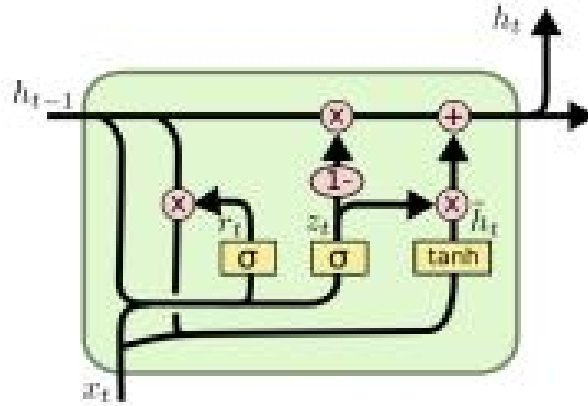


Figure 2.8: Schematic View of a GRU⁶.

2.2.2 Language Identification

Sometimes, determining the language of a text is the initial step of a broader natural language processing chain. Some languages even share homographs (such as room, which appears in both English and Dutch despite having different meanings in each language), which may confuse algorithms with the semantics of these particular words, necessitating the identification of the correct language for a given text prior to processing it further³⁹.

2.2.3 Intent Classification

Upon receiving a new message, the conversational agent has to be able to recognise the goal the user is trying to accomplish³⁹. This is commonly described as a multiclassification issue whose labels are the names of the possible user intentions. Techniques to handle this challenge vary from simple phrase extraction to Bayesian inference in order to determine the user's request based on various communications³⁹. LSTM networks have been previously known to work successfully in this area.

2.2.4 Knowledge Management

Without knowledge, an intelligent agent can only accomplish so much. In the 1980s, great progress was made in the discipline of knowledge engineering, which is concerned with allowing computers to manage knowledge³⁹. Earlier solutions often utilised an inference engine to alter data and generate new information using second and first order reasoning. It is a method for generating responses to incomplete inquiries, and they can typically be simply translated into API calls³⁹. Knowledge engineering is particularly beneficial for conversational agents, for example to answer basic questions about generic facts. Siri and Amazon Alexa acquire information from the Internet and other sources using knowledge inference techniques⁴⁶. Currently, API calls and optimised database queries are the most used methods for knowledge management. Even though more sophisticated methodologies inspired by graph-structured ontologies are occasionally utilised for knowledge bases⁴⁶.

2.2.5 Responses Generation

A conversational agent must possess the ability to reply to converse. Moreover, responses must be consistent with the context of the dialogue. This issue can be resolved by combining the efforts of two distinct modules: one that generates a list of candidate responses and the other that selects the most appropriate one or ranks them according to a particular metric³⁹. This subproblem has spawned two popular approaches: retrieval-based and generative-based methods³⁹. Using a huge database of candidate responses plus information from the user's message, retrieval-based algorithms identify the most appropriate response. This information may be a simple regular expression that searches for specific sentence structures, or it may be the result of a machine learning model. The primary benefit of this strategy is that the maintainers of the chatbot can control every response and so avoid incorrect replies³⁹. In contrast, generative-

based strategies rely on generative models to generate new responses without requiring a vast collection of instances. If the model is trained properly, it is straightforward to generate novel replies. However, as of now, the performance of generative-based techniques is insufficient relative to business limitations, and the industry remains sceptical of its potential³⁹. Recent attempts have been made to construct hybrid solutions in which responses are generated if they cannot be retrieved³⁹.

2.2.6 Performance Assessment

The performance evaluation and metrics used to determine the quality of a chatbot's behavior are one area for development in the field of conversational bots³⁹. In a paper titled "An Empirical Study of Unsupervised Evaluation Metrics for Dialogue Response Generation," the authors demonstrated how standard metrics in the field tend to be uncorrelated with intuitive human judgement, such as the BLEU and ROUGE scores⁴⁷. Speech and eloquence are essentially subjective, and it is consequently extremely difficult to measure them precisely, as is the case with any other subjective attribute of a system.

2.2.7 Design Principles

There are two approaches that can be used to develop a chatbot depending on the algorithms and techniques adopted: rule-based approach and machine learning approach³⁹.

Rule-Based

Using pattern-matching algorithms⁴⁸, a rule-based chatbot examines information and gives responses depending on a set of predetermined rules³⁹. Although the sophistication of pattern matching approaches varies, the fundamental concept remains the same. The user input is classed as a pattern, and the chatbot selects a predefined response by matching the pattern to a collection of predefined responses. Handcrafted are the pattern and response matching algorithms³⁹.

Numerous chatbots utilise pattern matching, with ELIZA, PARRY, and ALICE49 being early adopters of the technique. The advantage of the rule-based technique is its quickness, as it requires no in-depth text analysis. As the information is pre-determined by the creator, however, the responses are repetitive and lack flexibility and originality.

There are three languages most typically used for implementing rule-based chatbots.

- i. Artificial Intelligence Mark-up Language (AIML) is a derivation of Extensible Mark-up Language (XML)^{49,50}. ALICE was the first chatbot developed using the AIML programming language⁴⁹. AIML features a data object class known as AIML objects, which are responsible for modelling conversational patterns. Each item is comprised of two components known as subjects and categories⁴⁹. A subject is an optional top-level element with an associated set of categories and a name property. Categories are the most fundamental unit of knowledge and the chatbot's rules⁴⁹. Each category is comprised of two pieces referred to as pattern and template. The pattern is matched against the user's input to the template containing the chatbot's answer. AIML is one of the most often used languages for chatbot development due to its simplicity, flexibility, and good maintainability⁴⁹. The greatest drawbacks of AIML are that the developer must describe a pattern for every conceivable user input and that it is inefficient for large knowledge bases^{50,51}. Voici an illustration of the structure of an AIML object: Corresponding User Input Response to Input⁴⁹.

```
<category>
    <pattern>User Input</pattern>
    <template>
        Corresponding Response to Input
    <template>
</category>49.
```

- ii. **RiveScript**. is a scripting language based on lines that can be utilised to implement the Knowledge Base^{48,52}. RiveScript provides more built-in capabilities and tags than AIML, eliminating the need for the author to enter chatbot information in additional configuration files⁴⁹.
- iii. **ChatScript** is an open-source programming language for rule-based chatbot development. Through a technique known as dialogue flow scripting⁵³, it matches user inputs to outputs based on rules written by humans and stored in software scripts. ChatScript employs ideas, which are groups of words with the same meaning. It includes 2000 established concepts, and scripters can easily create their own concepts^{49,53}. ChatScript is a more difficult language than AIML and RiveScript, but it allows developers to combine rules in more complex ways⁴⁹.

2.2.8 Machine Learning

Recent advances in machine learning have enabled the creation of more intelligent chatbots⁵³. Using machine learning algorithms to extract information and generate responses, chatbots that employ machine learning strategies can improve from previous conversations⁴⁸. Chatbots based on machine learning necessitate a substantial training set. There are two sorts of models, retrieval and generative. In retrieval-based models, the ideal response is selected from a set of responses, whereas generative models employ deep learning techniques to produce the response⁴⁹.

2.2.9 Architecture

A general chatbot architecture consists of five main components, a User Interface component, a Natural Language Understanding (NLU) component, a Dialogue Management (DM) component,

a Backend component, and a Response Generation (RG) Component, as illustrated in the figure below⁴⁹.

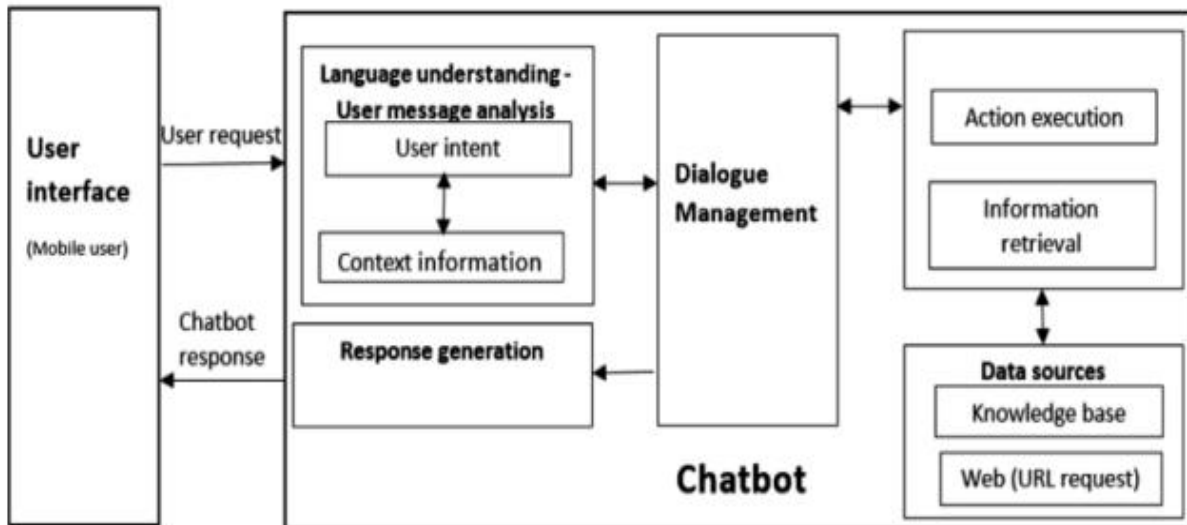


Figure 2.9: General Chatbot Architecture⁵⁵.

User Interface

The user interface permits users to converse and interact with a chatbot using messenger programmes such as Facebook Messenger, Cortana, and Slack⁴⁹. A chatbot's operation begins with a user's request. For voice-based conversational agents, an automated speech recognition (ASR) system will convert the user's speech input into text⁴⁹. A speech-based conversational agent is also equipped with a text-to-speech (TTS) system that turns text back into speech once a text response has been created. Text-to-speech entails two phases: text analysis and waveform generation. Text analysis entails normalising the text and doing phonetic (pronunciation) and prosodic (phrasing, pitch, loudness, tempo, and rhythm) analysis, whereas waveform synthesis is selecting the prerecorded speech that fits the most criteria⁴⁹.

Natural Language Understanding

After receiving the user's request, the system employs the NLU component to extract information from the input and provide a representation of its meaning that can be used in subsequent steps⁵⁵. NLU typically performs three tasks: categorization of dialogue acts, classification of intent, and slot filling⁴⁹. Classification of dialogue acts involves establishing the function of the user's input, or, more precisely, translating the user's utterance to a particular dialogue act type. The utterance may be categorised as a question, a statement, an offer, or another form of dialogue act. To better appreciate the user's request and determine an appropriate response, it is crucial to understand the current dialogue act⁴⁹. The intent classification identifies the user's principal objective. Intentions are generally domain-specific. For instance, a request could involve meal ordering, hotel reservations, weather forecasts, etc. The aim of an agent within the domain of hotel reservations can be to make, cancel, or modify a reservation, and similarly, the intent of an agent within the domain of food ordering can be to place, query, or modify an order⁴⁹. Slot filling is the final step of natural language understanding (NLU). Other relevant details are extracted by the agent, which, when paired with the dialogue act and the user's purpose, enable it to completely comprehend the user's request⁴⁹. There are various approaches for slot filling.

Regular Expression: is a tool for describing text patterns. Each pattern is specified by a sequence of characters and each character is either a regular character or a metacharacter that has a special meaning⁴⁹.

Tokenization entails dividing a text into units based on words, punctuation, or numbers⁴⁹. Since English words are typically separated by white space, tokenizing an English text should be quite

simple. However, there are some problematic instances in which the borders between words or phrases are unclear. Special cases include contracted elements (e.g., isn't), phrases (e.g., San Francisco), abbreviations (e.g., PhD.), and acronyms (e.g., AT&T)⁴⁹. Penn Treebank tokenization standardise a frequently used tokenization standard⁵⁶. According to the Penn Treebank tokenization standard, contractions and punctuation are split between tokens, although hyphenated words are maintained together. Using data to determine the tokens automatically is a more complex method of tokenizing text, and it is particularly beneficial when dealing with unfamiliar words⁴⁹. For example, bigger can be identified even if the training corpus contains only the terms "big," "little," and "smaller." Typically, this is accomplished by inducing a vocabulary from a row training corpus and then segmenting the text into tokens from the vocabulary⁴⁹. The vocabulary consists of subwords, such as "little" and "er" rather than "smaller." Byte-pair encoding, unigram language modelling, and WordPiece^{57,58} is the three most often utilised algorithms.

Text Normalization: refers to the process of transforming text to a standard format. Word normalisation is the process of identifying words with the same meaning but various spellings (e.g., G.R.A. and GRA). Case folding is the process of converting anything into lowercase letters.

Bag of Words: the vector space model, also regarded as one of the simplest ways to input analysis, is also known as the vector space model. We count the frequency of each word but disregard syntactic and word order information. This can be achieved by normalising the text and removing stop words, or words that do not contribute to the meaning of the sentence⁴⁹. When the meaning of a document depends on linguistic information, such as grammar or word order, the

bag-of-words technique is inadequate. For instance, "John is taller than Mike" and "Mark is taller than Luke" have the same representation but have distinct meanings⁵⁹.

Latent Semantic Analysis (LSA): or Latent Semantic Indexing, does not account for any linguistic knowledge. However, LSA compares the meaning behind the words rather than the words themselves, and keywords that frequently appear in the same context are grouped together⁴⁹. To determine word similarity, a matrix is created in which each row represents a term, each column represents a document, and each cell indicates the frequency of the word in document⁴⁹. The matrix is subsequently transformed by applying Singular Value Decomposition to minimise its dimensionality⁴⁹.

Part-of-speech (POS) Tagging: entails marking up each word in a text with a tag indicating its syntactic function, such as whether it is a verb, noun, pronoun, conjunction, etc. POS tagging is particularly beneficial when we encounter terms with several meanings and their application in a sentence may be ambiguous⁴⁹. For instance, the word "ring" can be either a noun or a verb, but in the sentence "ring a bell," it should be classified as a verb. Using a POS tag in such instances can eliminate the ambiguity⁵⁹.

Dialogue Management: The Dialogue Management component handles information flowing from other components and is responsible for controlling and modifying the context of discussions as well as governing the chatbot's activities⁵⁹. DM coordinates with other modules and encompasses the speech application's logic. This indicates that it is a crucial component of the chatbot architecture, and we must make deliberate and considered design decisions⁵⁹. Unfortunately, building a resilient DM strategy is not a simple task, as it is difficult to predict which system behaviour would result in high user satisfaction. In this paragraph, we explore two

design issues of DM that arise frequently: interaction methods and confirmation strategy selection. A dialogue system's interaction approach determines who controls the discourse. The dialogue might be directed by the user, the system, or a combination of both. When the interaction is user-directed, the user takes the lead, and the system simply responds to the user's questions and orders. When the interaction is system-directed, the system takes the lead, and the user simply responds to its questions. When the dialogue is mixed-initiative, both the user and the system are able to take the initiative⁵⁹. The DM monitors the condition of the discussion or the system's conversational behaviour, which is based on the interaction strategy it adopts⁴⁹.

Backend Chatbots: Retrieve from the Backend the information required to complete the relevant tasks, and then transmit the message to the Dialogue Management Component and the Response Generation Component⁴⁸. The handmade rules must be stored in a Knowledge Base (KB) for rule-based chatbots. To ensure the robustness of the chatbot, the Knowledge Base rules should be as numerous and comprehensive as feasible⁶⁰. A chatbot can utilise an RDB to recall earlier talks. Taking into consideration prior information enables the chatbot to interact with greater consistency, precision, and dependability⁶¹. The development of the KB is crucial and required, but it can be time-consuming and stressful because to the manual labour involved. In order to solve this obstacle, developers devised a method that can automatically generate a new KB from an existing chatbot's KB⁶².

Frequently, rule-based chatbots direct users and populate the KB with their responses⁶³. Recently, there has been a tendency toward retaining the information contained in digitally saved texts on the Internet and other online sources. Large data repositories that store data in a machine-readable and accessible format⁴⁹. Google's Knowledge Graph, DBpedia, Freebase, and Wolfram Alpha are some examples of these knowledge bases⁴⁹.

Response Generation: Once the required information has been gathered, the conversation system determines the response's content and the optimal way to convey it⁴⁹. The Response Generation component is responsible for creating responses in a format that is easily understood by the end user. Five steps of processing comprise the RG pipeline: signal analysis, data interpretation, document planning, microplanning, and implementation⁵⁹. Analysis of signals and interpretation of data are preliminary steps. These two stages are traversed by inputs in the form of data, such as numerical data in weather forecasting. Using pattern matching algorithms, the signal analysis finds data patterns and trends⁵⁹. Data interpretation employs symbolic reasoning techniques based on subject expertise to comprehend complicated messages and message relationships. Document planning can be subdivided into two more phases: content determination and discourse planning⁴⁹. Determining content entails deciding what information must be supplied to the user. The system sends only a portion of the obtained data, and it may communicate the data in phases. Whenever necessary, determining content also includes filtering and summarising the information. The objective of discourse planning is to organise and structure the text to facilitate human comprehension⁴⁹. Microplanning, often known as sentence planning, is the process of composing the information-delivery sentences. It includes three tasks: referential expressions, aggregation, and lexical selection⁴⁹.

2.3 Review of Related Work

The review of the literature reveals that several authors have contributed significantly to the development of chatbot.

In a paper titled design and deployment of a chatbot in the context of customer service³⁹. This paper gives a practical case study of a chatbot solution for the GAMING1 organisation. The

author explained the theory underlying the tactics employed. Specifically, gated recurrent unit neural networks and other techniques of deep learning are described. Then, a list of the difficulties resolved by the chatbot is written down. The author then proposes and explains a scalable software architecture for the chatbot. Models for requesting a human operator are presented alongside a comprehensive investigation of several neural network structures for user intent classification. It was determined that gated recurrent units were the most successful for classification, although simpler models performed pretty well for human operator requesters. Finally, performance metrics for the chatbot's many submodules are summarised. However, because performance measurements are difficult to interpret for dialogue systems, several practical test cases are shown to demonstrate that the bot's behaviour is more than sufficient despite certain performance metrics remaining unsatisfactory.

In another related study on the development of chatbots in higher education⁶⁴. This research uses the Facebook Page of an academic programme to highlight the potential benefits of implementing chatbots. The end-to-end process of developing this chatbot utilising Google Dialogflow as the NLU platform, and Facebook Messenger as the platform interface is also explained. As a representative sample, the chat logs were mined for a total of 807 sentences from 125 people, from which 33 intents can be derived. The suggested chatbot obtains a satisfactory precision, recall, and F1-score of 0.98, 0.888, and 0.897%, respectively. The use of chatbots resolved issues for both page managers and end users.

In an Innovative Approach to Arabic Chatbot, Google Colab and the Internet of Things are utilised⁶⁵. In this research, a new Internet of Things-enhanced strategy for the building of

effective Arabic chatbots (IOT). Using Google Colab and the Python Chatterbot module, an experiment was conducted to develop and deploy an Arabic Chabot for an IOT-based data center.

In a study examining the effect of social virtual presence agents and a content-based product recommendation system on the online purchase intent of customers⁶⁶. The purpose of this study is to examine the effects of utilising two distinct chatbot social virtual presences interactions: with a fully pre-recorded computed personification agent versus with a pre-recorded human social virtual agent; and also, to comprehend how having a customised content-based product recommendation system can influence the consumers' purchase intention in an online shopping framework. The core data were processed using a PLS-SEM model on an online platform that simulated a potential virtual store encounter. While assisting shoppers, the Human Social Virtual Presence Agent has a greater model positive effect on Intellectual stimulation and Hedonic Benefits than a computer-generated personification Agent. This could be explained by the fact that participants experienced Agent's computer-generated images and sounds as emotionally unsettling. Additionally, the presence of a recommendation system influences the purchase intent of customers in a positive manner, compared to the absence of a recommendation system. This study demonstrates how important social contacts are for customers, particularly when conducted by a human, and how the recommendation system influences customers' buy intent.

In a related work, Intelligent Chatbot Adapted from Question-and-Answer System Using RNN-LSTM Model was developed⁶⁷. The chatbot is created to retain data gathered via a question-and-answer system, which can then be utilised in the Python application. This application will utilise the Cornell Movie Dialog Corpus, a dataset containing a corpus including a vast collection of fictitious talks derived from film scripts and rich in metadata. The Python chatbot application can

employ a variety of models, but the LSTM model is used, specifically in this software. The output of the chatbot programme employing the LSTM model consists of accuracy as well as a data set that corresponds to the information entered by the user in the chatbot dialogue box. The choice of models that can be used is dependent on data that can affect programme performance, with the goal of the programme deciding the high or low level of accuracy that will be generated from the program's findings, which can be a significant role in choosing the chosen model. Based on the incorporation of the LSTM model into the chatbot, it can be concluded that Parameter Pair 1 (size layer 512, num layers 2, embedded size 256, learning rate 0.001, batch size 32, epoch 20) from File 3 is the optimal parameter pair for the LSTM Chatbot with an average accuracy of 0.994869.

In the Design and Implementation of a Chatbot for Speakers of the Kurdish Language Using the Chatfuel Platform⁶⁸. The design and execution of a chatbot are presented to assist Kurdish speakers in using text-based internet discussions as an alternative to contacting human operators directly. The software agent based on natural language processing is implemented utilising the Chatfuel platform. Chatfuel employs artificial intelligence to communicate with humans by replicating human conversations via voice instructions or text messages. The proposed chatbot is tested on an electronic tourist guide that assists tourists to the religious sites in the Iraqi Kurdistan mountain hamlet of Barzanja. Three hundred questions and responses are used to conduct the case study. One hundred participants took part in this study. The participant asks a question, and the bot responds if it recognises the question; otherwise, it responds with a default answer and instructions on how to utilise the system correctly. These experiments' data are collected, examined, and difficulties with the Kurdish language are identified. Creating software agents for parsing Kurdish texts presents numerous obstacles. Natural language processing has

not yet been applied to Kurdish texts (NLP). In addition, the Kurdish typeface disorder and lack of conventional keyboards and writing styles make it difficult to process Kurdish text. In addition, the Kurdish language consists of a range of dialects with distinct writing styles.

To design for university a conversational question-and-answer chatbot⁶⁸. The author suggests the development of "A Conversational Question and Answer Chatbot for University" to reduce the amount of manual labour performed by university marketing and promotion teams in response to information queries. A review of existing chatbot systems led to the adoption of the AI Chatbot Development Methodology and the selection of the Nave Bayes method as the NLP Engine for the chatbot development implementation. The deployment of the chatbot revealed that it could manage the majority of discussions within a restricted scope. Further stabilisation is required to strengthen the chatbot's NLP Engine so that it can handle all conceivable user interactions.

In a study connected to the design and execution of xiaoice, an empathic social chatbot⁷⁰. This article discusses the creation of Microsoft XiaoIce, the world's most popular social chatbot. XiaoIce is particularly built as a companion with artificial intelligence and an emotional connection to fulfil the human need for communication, affection, and social belonging. In system design, they considered both intellectual quotient and emotional quotient, portrayed human-machine social chat as decision-making rather than Markov Decision Processes, and optimised XiaoIce for long-term user engagement, as measured by anticipated Conversation-turns Per Session (CPS). The system architecture and main components, including the dialogue manager, core chat, skills, and an empathic computing module, are described in detail. Throughout lengthy talks, we demonstrate how XiaoIce dynamically recognises human emotions and states, comprehends user intent, and responds to user demands. Since its inception in 2014,

XiaoIce has communicated with more than 660 million active users and established long-term relationships with a significant number of them. Analysis of large-scale online logs reveals that XiaoIce's average CPS of 23 is significantly higher than that of other chatbots and even human conversations.

In the development of IntelliBot⁷¹. This study investigates the response generation strategies of existing chatbots and identifies their deficiencies from the perspective of engaging in a discourse with a user. It then offers a domain-specific chatbot called IntelliBot, which is a dialogue-based chatbot that generates responses using several methodologies. IntelliBot possesses domain-specific expertise as a result of its training on two datasets, namely the Cornell movie dialogue dataset and a custom-built insurance dataset. The performance of IntelliBot was then validated and compared to that of three other chatbots described in the literature: RootyAI, ChatterBot, and DeepQA. The outcomes illustrate IntelliBot's superiority in engaging the user and providing a comprehensive response in the insurance domain.

In a similar study on Cognitive chatbots for personalised context-based customer service⁷². It was determined that the chatbots were effective. This study addresses cognitive chatbot-based personalised contextual customer support. The primary data acquired from 300 B2B corporate respondents is analysed using a quantitative research approach. The study contributes to the little research on chatbots and offers customer service enhancements. The findings indicate that customers place a high value on real-time information regarding the dependability and accessibility of products/services. Customers enjoy a seamless experience due to the automated responses to repetitive questions about recurring concerns. This study provides substantial theoretical contributions to the chatbot literature by merging two models into a simplified model

and demonstrating that trust influences the propensity to employ intelligent chatbots, which promote automation.

In the design of an intelligent knowledge-based customer service chatbot⁷³. This study presents an intelligent knowledge-based conversational agent system architecture for e-commerce sales and marketing customer service support. In a leading manufacturer of intimate apparel for women, a prototype implementation of a chatbot for customer care is reported. The suggested system includes several cutting-edge technologies, such as web crawling, natural language processing, knowledge bases, and artificial intelligence. In this work, a prototype system is constructed in an actual environment. The findings of the examination of the system's prototype are satisfactory and support the claim that the system is effective.

Further, in the development of university chatbot⁷⁴. The idea involves interaction between users and a chatbot that can be accessible at any time and from any location. Using a few easy language conversions, the chatbot may be simply integrated into the websites of any university or college. The chatbot gives numerous universities or college-related and student-specific pieces of information. The chatbot is accessible to anyone with access to the university's website. The project employs Artificial Intelligence and Machine Learning concepts. Chatbots are created using the PHP programming language. The user can ask university-related queries, which are subsequently fed into an algorithm that processes the message and shows the associated response to the user. The Project GUI resembles a Messaging Program.

To develop artificial intelligent marketing chatbot⁷⁵. This article examines the chatbot as an artificial intelligence marketing tool, its current implementation, and its future potential in the industry. Sixty respondents were surveyed regarding their expectations, behaviours, and habits

when utilising different communication channels, with a focus on chatbots, their advantages and disadvantages in comparison to other communication channels. The results revealed that the greatest benefit of employing chatbots in marketing services was when delivering simple, quickly accessed information, but also revealed respondents' concern that chatbots might provide them with incorrect information. Organizations might consider utilising chatbots if they face difficulties in communicating with customers and if they want to stay up with the lifestyles of an increasing number of consumers.

To develop a chatbot for college students, advisement⁷⁶. This paper outlines the design and development of EASElective, a conversational agent for course selection. EASElective is intended to supplement existing academic advising services with an online natural language interactive interface that would facilitate conversations on issues ranging from basic official course data to informal student comments. EASElective's key design components are intent detection, conversation management procedures, dialogue design, sustainable opinion gathering and analysis from students, and course information management. The article also offers a study of EASElective's perceived effectiveness.

In a similar analysis of Implementation of current state-of-the-art technology for obtaining a functional Dialogue Agent⁷⁷. The authors present the technique used to construct a Polish-speaking bot that can book services within the existing booking system Reservis by interacting with a user. A form of expert system consisting of a task-oriented discussion system has been developed. In addition to fundamental functionality, two more objectives were established: to make human-computer interaction as natural as possible and to avoid implicitly limiting user choice (and therefore showing only part of available results). In contrast to the majority of

commercial conversational agents, the presented dialogue manager is purely goal-oriented and designed to accomplish a complex task by sustaining a multi-turn conversation.

Using Ontology and Transfer Learning in the design and development of a Language Learning Chatbot with XAI⁷⁸. The authors suggested a chatbot for English language acquisition based on transfer learning, whose output generated by GPT-2 may be justified by a related ontology network rooted in a fine-tuning dataset. The designers created three levels for systematically learning English, including a phonetics level for speech recognition and pronunciation correction, a semantic level for domain-specific conversation, and the simulation of free-style conversation in English - the highest level of language chatbot communication as a free-style conversation agent. Following the notion of XAI (Explainable Artificial Intelligence) to show the connections of neural network in bionics and explain the output sentence from language model, we build the ontology graph to explain the performance of free-style discussion as an academic contribution. In terms of implementation, our Language Learning agent incorporated the mini-program in WeChat as the front-end and fine-tuned the GPT-2 model of transfer learning as the back-end in order to interpret the responses by ontology graph.

In a study examining the prevalence of chatbots in the food service delivery industry and their effect on the customer experience⁷⁸. The study investigated the consequences of chatbots' engagement styles for younger users of conversational online food delivery (OFD) services. This research investigates whether the conversational design of chatbots can influence social, affective, and behavioural effects. The study employs an experimental design to examine the impact of a social- versus task-oriented interaction style chatbot on the amount of social presence and trust (social outcomes), reported enjoyment (affective outcomes), and desire to utilise the

conversational OFD service in the future (behavioral intent outcome). The encounter with a chatbot configured with a social-oriented interaction style boosted users' impression of social presence and enjoyment but had no direct and meaningful influence on trust and intent to use.

In the design of COVIBOT: An Intelligent Chatbot for Assistance and E-Awareness During the COVID-19 Pandemic⁸⁰. This study intends to create and implement a smart chatbot on the topic of COVID-19, named COVIBOT, that will be useful in Saudi Arabia. COVIBOT is a generative-based contextual chatbot that is constructed using machine learning APIs provided by Azure Cognitive Services. COVIBOT is available in both English and Arabic editions. COVIBOT use cases are tested and validated using a scenario-based methodology.

To explore, examine, and compare the psychotherapy applicability and limitations of contemporary chatbots⁸¹. The poll identifies a number of tasks that will be required in future therapeutic chatbots. Five standard and cutting-edge psychotherapy chatbots were chosen after a search of over 1,200 relevant articles in public databases. Most contemporary psychotherapy chatbots generate dialogue using retrieval-based strategies. Certain psychotherapy chatbots employ psychological ideas, such as cognitive behaviour therapy, to address specific psychological issues. The evaluations indicate that chatbots can distinguish distinct unpleasant emotions and provide relatively suitable responses. Randomized controlled trials demonstrate the usefulness of therapeutic chatbots for some individuals with mental health conditions. Compared to actual psychologists, psychotherapy chatbots provide a number of advantages, including accessibility regardless of time or location.

A distributed data base model for social networking and data analytics distributed serverless chatbot networks using mobile agents⁸². Mobile agents are used to construct a distributed,

decentralised, serverless dialogue robot network that enables ad hoc communication between humans and machines (networks) as well as between human groups through the chatbot network (supporting personalised and mass communication). Using a distributed knowledge and data base approach, chatbot networks attempt to expand the communication and social interaction capabilities of people, especially in mobile situations. Moreover, the chatbot network is a sensor data gathering and data aggregator system that enables crowd-based analytics on a massive scale. A first proof-of-concept demonstrator is presented that identifies the difficulties posed by self-organizing, distributed chatbot networks in resource-constrained mobile networks. A hybrid chatbot multi-agent architecture that enables scalable dispersed and adaptive communicating chatbot networks is the novelty of this work.

In a similar study, a conceptual redesign of the Information Retrieval (IR) chatbot architecture is proposed to minimise the necessity for huge Question-Answer (QA) pair datasets in machine learning training and knowledge base development of chatbots⁸³. The authors described Ans2Q: a Neural Network model for question type approximation, and HR6: an IR score ranking computation based on Ans2Q output, within ten recommended framework components. These two elements represent the fundamental distinction between the proposed framework and others. In addition to describing the process flow, we present numerous formulas that can possibly be used to build this framework. With this framework, we intend to create a tool that may be used to develop chatbots for nearby domains with less expertise and no easily available QA pair datasets. In the development of AI-Powered Health Chatbots⁸⁴. This paper proposes a general Architecture of an AI-Powered Health Chatbot with Four components to achieve the two goals, which integrates the dialogue and communication part in natural language understanding (NLU)

and natural language generation (NLG), and the expert part based on deep learning whose function is to provide an appropriate response from pre-formatted data.

Likewise, in the design and implementation of an Intelligent English Learning Chabot based on Transfer Learning Technology, similar considerations were made⁸⁵. The author presented a Transfer-based English Language learning chatbot with three learning system levels for practical application, which integrates Google's recognition service and GPT-2 Open AI with dialogue tasks in NLU and NLG at a WeChat mini-program. From an operational standpoint, three layers for methodically learning languages were developed: phonetics, semantics, and an English "freestyle conversation" simulation. First level consists of correcting voice recognition pronunciation and learning sentence syntax. The second level is a converse-specific domain, and the third level is a language chatbot communicating as a freestyle conversation agent. The Language Learning agent integrates into a WeChat mini-program to design three user interface levels and to refine transfer learning GPT-2 as the back-end language model used to generate responses for users. With the combination of the two parts concerning operation and implementation, based on the Neural Network model of transfer learning technology, various users test the system with an open-domain topic, acquiring good communication experience, and demonstrating that it is ready to be used in an industrial setting.

To examine the elements that influence consumer happiness in voice commerce and e-commerce⁸⁹. Four elements were explored in the study, leading the author to generate four hypotheses. (These variables include Recommendation Complexity, Personalization of Recommendations, Convenience, and Transaction Process Efficiency. A survey was conducted with 178 consumers and used structural equation modeling for statistical hypotheses testing. The results showed that consumers have higher expectations in convenience for voice commerce than

they have for e-commerce. The result also showed that transaction process efficiency significantly influences satisfaction in voice commerce, but not in e-commerce.

Additionally, Ontology-based chatbots can be installed on e-commerce websites⁹⁰. The authors believe that the ontology-based chatbot will provide the user with substantive responses and a more natural and engaging dialogue. Pattern-based chatbots, unlike ontology-based chatbots, have preprogrammed responses, making their dialogue unnatural⁹⁰.

Another study explored how chatbots function in an academic setting by creating an academic chatbot to aid academic advising bodies in Nigeria⁹¹. A research also created a chatbot that works as a tutor for software engineers, the learners in this instance⁹². In a different study of a similar nature, the authors developed a chatbot as a historical figure, with the chatbot containing a great deal of knowledge about the life and personality of the simulated person in order to operate in line with its required character.

The use of chatbots in healthcare includes support, motivation, patient coaching, and administrative job organization⁹³. Concerns exist, however, regarding the inability of chatbots to empathise with patients and provide competent medical advice in specialised fields. Additionally, chatbots have been implemented in the e-commerce sector. A study emphasised the practical benefits of employing AI in e-commerce, citing the case of Amazon, where the use of AI has led to increased productivity, an improvement in customers' online shopping skills, and a favourable influence on the company's revenue⁹⁴. Scholar similarly shares insights into how chatbots have changed e-commerce marketing⁹⁵. In the e-commerce industry, chatbots have the highest level of client approval, according to the research. In addition, chatbots are anticipated to be the digital imitator of humans in 2050⁹⁵. The author argues that, considering these worrying statistics, the conversational level of chatbots must be enhanced by boosting the machine learning level

capability of the bots and developing chatbots with an all-in-one technology, such as NLP, ontology-based, and pattern-based chatbots.

In addition, a study deploying a ChatterBot as a negotiator for e-commerce websites further discussion on chatbots⁹⁶. A distributed chatbot system for supply chain management was also built as part of the same endeavour⁹⁷. Another study elucidated how an e-commerce chatbot may create autonomous responses to clients' product and service inquiries⁹⁸. Although a business may run 24 hours a day, seven days a week, the writers stated that it is improbable that its staff will be available at all times. SuperAgent is an open-domain chatbot that can be integrated in web browsers like as Microsoft Edge and Google Chrome, according to a study⁹⁹. In the study, SuperAgent is depicted as a customer care chatbot that makes use of massive size and freely available e-commerce datasets.

In a separate study, the authors created and integrated a chatbot component into the Customer Relationship Management (CRM) system of a medium-sized Austrian firm¹⁰⁰. Also proposed is a chat-based automated system for internet purchasing called "CartBot." It is often referred to as a personalised assistant that recognises the demands of its consumers based on their likes and preferences¹⁰¹. In another study, the use of chatbots as sales assistants was demonstrated by constructing a chatbot in an e-commerce system that conducts basic discussions regarding the remaining stock of each product, orders, and payments¹⁰².

2.4 Chapter Summary and Gap in Literature

This chapter was organised into four sub-headings - conceptual review, theoretical review/framework, review of empirical studies related to the research topic and conceptual model. The conceptual review explained in depth the concepts of the study. These concepts are -

human computer interaction, components of chatbot, capabilities and metrics, uses of chatbot, history. It also richly gave insights into sub-concepts such as different types of chatbot which includes ELIZA, Elizabeth, and ALICE.

In the methodological review, the different techniques for the development of chatbots used in this study were fully explained. They are common deep learning techniques such as word embedding, Recurrent Neural Networks. Also, language identification, intent classification, knowledge management, responses generation, design principles, machine learning and architecture were discussed.

In the review of empirical studies, different authors contribution to the development of chatbot were presented. Also, closely related studies on automated customer support system (chatbot) were also presented. The studies show that many empirical research works like the topic under study have been carried out. However, past empirical studies using other algorithms showed lower accuracy.

Also, these solutions did not generate a google script to translate from any language to English for the bot to understand and send the response in the preferred language of the user. To the best of researcher's knowledge, there is a paucity of literature on using AIML file for FAQ questions. The empirical studies reviewed therefore shows that studies are lacking in the subject matter which identifies a gap in literature that needs to be filled. Therefore, this research proposes to develop a more intelligent chatbot that will automate certain aspects of customer support and offload some of the workload from support service agents, thereby lowering the volume of customer complaints related to support team response delays and capable of answering customer

complaints thereby reducing turnaround time, delays in transaction, time waste, and financial loss.

Do Not Copy, Lead City University, Nigeria

Endnotes

1. F Peters. Master thesis: *Design and implementation of a chatbot in the context of customer support*. Masters Dissertation.2018.
2. A Truong, A Walters, J Goodsitt, K Hines, CB Bruss, & R Farivar. *Towards automated machine learning: Evaluation and comparison of AutoML approaches and tools*. In2019 IEEE 31st international conference on tools with artificial intelligence (ICTAI) 2019 Nov 4 (pp. 1471-1479). IEEE.
3. J Barber, X Yu, LK Zamore, J Lin, V Jazayeri, S Erlich, T Savor, & M Stumm. *Bladerunner: Stream Processing at scale for a live view of backend data mutations at the edge*. InProceedings of the ACM SIGOPS 28th Symposium on Operating Systems Principles 2021 Oct 26 (pp. 708-723).
4. E Merdivan, D Singh, S Hanke, & A Holzinger. *Dialogue systems for intelligent human computer interactions*. Electronic Notes in Theoretical Computer Science. 2019 May 4;343:57-71.
5. T Schachner, R Keller, & F von Wangenheim. *Artificial intelligence-based conversational agents for chronic conditions: systematic literature review*. **Journal of medical Internet research**. 2020 Sep 14;22(9):e20701.
6. Y Yun, D Ma, & M Yang. *Human-computer interaction-based decision support system with applications in data mining*. Future Generation Computer Systems. 2021 Jan 1;114:285-9
7. AA Weißensteiner. *Chatbots as an approach for a faster enquiry handling process in the service industry*. Signature. 2018;12(04).
8. P Ifinedo, J Pyke, & A Anwar. *Business undergraduates' perceived use outcomes of Moodle in a blended learning environment: The roles of usability factors and external support*. Telematics and Informatics. 2018 Apr 1;35(1):93-102.
9. S Schanke, G Burtch, & G Ray. *Estimating the impact of "humanizing" customer service chatbots*. Information Systems Research. 2021 Sep;32(3):736-51.
10. AJ Kockx. *Development and Evaluation of a Diagnosis and Triage Healthcare Chatbot*. (Master's thesis).2018

11. SJ Español, R Fronda Jr, MR Villarica, & GL Intal. *SHOP BUDDY Enhanced Automated Chatbot for Any E-commerce Website with Data Analytics Functionalities*.
12. L Wang, D Wang, F Tian, Z Peng, X Fan, Z Zhang, M Yu, X Ma, & H Wang. *Cass: Towards building a social-support chatbot for online health community*. *Proceedings of the ACM on Human-Computer Interaction*. 2021 Apr 22;5(CSCW1):1-31.
13. P Smutny, & P Schreiberova. *Chatbots for learning: A review of educational chatbots for the Facebook Messenger*. *Computers & Education*. 2020 Jul 1;151:103862.
14. A Følstad, & M Skjuve. *Chatbots for customer service: user experience and motivation*. In *Proceedings of the 1st international conference on conversational user interfaces 2019* Aug 22 (pp. 1-9).
15. SJ Español, R Fronda Jr, MR Villarica, & GL Intal. *SHOP BUDDY Enhanced Automated Chatbot for Any E-commerce Website with Data Analytics Functionalities*.
16. P Singh, & A Kaur. *A systematic review of artificial intelligence in agriculture*. *Deep Learning for Sustainable Agriculture*. 2022 Jan 1:57-80.
17. O Allal-Chérif, AY Aranega, & RC Sánchez. *Intelligent recruitment: How to identify, select, and retain talents from around the world using artificial intelligence*. *Technological Forecasting and Social Change*. 2021 Aug 1;169:120822.
18. C Akcora. *The effect of chatbots tone of voice on trust and customer satisfaction* (Master's thesis, University of Twente).
19. A Ghimire, S Thapa, AK Jha, S Adhikari, & A Kumar. *Accelerating business growth with big data and artificial intelligence*. In *2020 Fourth International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC) 2020 Oct 7* (pp. 441-448). IEEE.
20. M De Gennaro, EG Krumhuber, & G Lucas. *Effectiveness of an empathic chatbot in combating adverse effects of social exclusion on mood*. *Frontiers in psychology*. 2020 Jan 23;10:3061.
21. KJ Hayward, & MM Maas. *Artificial intelligence and crime: A primer for criminologists*. *Crime, Media, Culture*. 2021 Aug;17(2):209-33.

22. H Rainie, JQ Anderson, & J Albright. *The future of free speech, trolls, anonymity and fake news online*. Washington, DC: Pew Research Center; 2017 Mar 29.
23. S Melián-González, D Gutiérrez-Taño, & J Bulchand-Gidumal. *Predicting the intentions to use chatbots for travel and tourism*. *Current Issues in Tourism*. 2021 Jan 17;24(2):192-210.
24. J Hill, WR Ford, & IG Ferreras. *Real conversations with artificial intelligence: A comparison between human-human online conversations and human-chatbot conversations*. *Computers in human behavior*. 2015 Aug 1;49:245-50.
25. W Maroengsit, T Piyakulpinyo, K Phonyiam, S Pongnumkul, P Chaovalit, & T Theeramunkong. A survey on evaluation methods for chatbots. In *Proceedings of the 2019 7th International Conference on Information and Education Technology 2019 Mar 29* (pp. 111-119).
26. K Kretzschmar, H Tyroll, G Pavarini, A Manzini, & I Singh, NeurOx Young People's Advisory Group. *Can your phone be your therapist? Young people's ethical perspectives on the use of fully automated conversational agents (chatbots) in mental health support*. *Biomedical informatics insights*. 2019 Feb;11:1178222619829083.
27. MH Huang, & RT Rust. *Engaged to a robot? The role of AI in service*. **Journal of Service Research**. 2021 Feb;24(1):30-41.
28. C Rutschi, & J Dibbern. *Towards an Understanding of Scaling the Software Robot Implementation*. In *Information Systems Outsourcing 2020* (pp. 453-466). Springer, Cham.
29. R Richer, N Zhao, BM Eskofier, & JA Paradiso. *Exploring smart agents for the interaction with multimodal mediated environments*. *Multimodal Technologies and Interaction*. 2020 Jun;4(2):27.
30. M Nuruzzaman, & OK Hussain. *A survey on chatbot implementation in customer service industry through deep neural networks*. In *2018 IEEE 15th International Conference on e-Business Engineering (ICEBE) 2018 Oct 12* (pp. 54-61). IEEE.
31. LF D'Haro, S Kim, KH Yeo, R Jiang, AI Niculescu, Banchs RE, & Li H. *CLARA: a multifunctional virtual agent for conference support and touristic information*. In *Natural language dialog systems and intelligent assistants 2015* (pp. 233-239). Springer, Cham.

32. R Law, R Leung, A Lo, D Leung, & LH Fong. *Distribution channel in hospitality and tourism: Revisiting disintermediation from the perspectives of hotels and travel agencies*. **International Journal of Contemporary Hospitality Management**. 2015 Apr 13.
33. S Melián-González, D Gutiérrez-Taño, & J Bulchand-Gidumal. *Predicting the intentions to use chatbots for travel and tourism*. *Current Issues in Tourism*. 2021 Jan 17;24(2):192-210.
34. G Chrysovelidis. *Designing a chatbot for tourism*. Master of Science (MSc) Dissertation in E-Business and Digital Marketing November 2019
35. P Anita Nathania, & VY Yulia. *Android Based Chatbot and Mobile Application for Tour and Travel Company*. *IJCT*. 2018:21.
36. S Natale. *Deceitful media: Artificial intelligence and social life after the Turing test*. Oxford University Press, USA; 2021 Mar 17.
37. GK Ahirwar. *Chatterbot: technologies, tools and applications*. In *High Performance Vision Intelligence 2020* (pp. 203-213). Springer, Singapore.
38. N A Al-Madi, K.A Maria, M.A Al-Madi, & M.A Alia. and Maria, & E.A., 2021, July. *An intelligent Arabic chatbot system proposed framework*. In *2021 International Conference on Information Technology (ICIT)* (pp. 592-597). IEEE
39. Peters F. *Design and implementation of a chatbot in the context of customer support*. Master thesis: 2018
40. S Thirumuruganathan, N Tang, MOuzzani, & A Doan A. *Data curation with deep learning [vision]*. arXiv preprint arXiv:1803.01384. 2018 Mar 4.
41. P Kumari, & D Toshniwal. *Long short term memory–convolutional neural network based deep hybrid approach for solar irradiance forecasting*. *Applied Energy*. 2021 Aug 1;295:117061.
42. XH Le, HV Ho, G Lee, & S Jung. *Application of long short-term memory (LSTM) neural network for flood forecasting*. *Water*. 2019 Jul;11(7):1387.
43. Olah C. *Understanding LSTM Networks*. [(accessed on May 17, 2022)].
44. R Rana. *Gated recurrent unit (GRU) for emotion classification from noisy speech*. arXiv preprint arXiv:1612.07778. 2016 Dec 13.

45. A Ram, R Prasad, C Khatri, A Venkatesh, R Gabriel, Q Liu, J Nunn, B Hedayatnia, M Cheng, A Nagar, & E King. *Conversational ai: The science behind the alexa prize*. arXiv preprint arXiv:1801.03604. 2018 Jan 11.
46. A Ram, R Prasad, C Khatri, A Venkatesh, R Gabriel, Q Liu, J Nunn, Hedayatnia B, M Cheng, A Nagar, & E King,. *Conversational ai: The science behind the alexa prize*. arXiv preprint arXiv:1801.03604. 2018 Jan 11.
47. CW Liu, R Lowe, IV Serban, M Noseworthy, L Charlin, & J Pineau. How not to evaluate your dialogue system: *An empirical study of unsupervised evaluation metrics for dialogue response generation*. arXiv preprint arXiv:1603.08023. 2016 Mar 25.
48. E Adamopoulou, L Moussiades. *An overview of chatbot technology*. InIFIP International Conference on Artificial Intelligence Applications and Innovations 2020 Jun 5 (pp. 373-383). Springer, Cham.
49. X Huang, A CIS. *Chatbot: Design, Architecture, And Applications*. ASCS CIS 498 Senior Capstone Thesis University of Pennsylvania, School of Engineering and Applied Science May 3, 2021
50. S Pardeshi, S Ovhal, P Shinde, M Bansode, & A Birajdar. *A survey on Different Algorithms used in Chatbot*. **International Research Journal of Engineering and Technology (IRJET)**. 2020;7:6092-8
51. A Trivedi, V Gor, & Z Thakkar. *Chatbot generation and integration: A review*. **International Journal of Advance Research, Ideas and Innovations in Technology**. 2019;5(2):1308-11.
52. NP Naik. Performance Measurement of Natural Dialog System by Analyzing the Conversation. In*Designing User Interfaces With a Data Science Approach 2022* (pp. 180-209). IGI Global
53. S Hussain, O Ameri Sianaki, & N Ababneh. *A survey on conversational agents/chatbots classification and design techniques*. In*Workshops of the International Conference on Advanced Information Networking and Applications 2019* Mar 27 (pp. 946-956). Springer, Cham.
54. I Hsu, & JD Yu. *A medical Chatbot using machine learning and natural language understanding*. *Multimedia Tools and Applications*. 2022 Mar 18:1-23.

55. Y Zhang, L Xu, A Mendoza, G Yang, P Chinprutthiwong, & G Gu. *Life after speech recognition: Fuzzing semantic misinterpretation for voice assistant applications*. In Proc. of the Network and Distributed System Security Symposium (NDSS'19) 2019 Jan.
56. A Pinto, OH Gonçalo, & AA Oliveira. *Comparing the performance of different NLP toolkits in formal and social media text*. In 5th Symposium on Languages, Applications and Technologies (SLATE'16) 2016. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik.
57. T Kudo, & J Richardson. *Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing*. arXiv preprint arXiv:1808.06226. 2018 Aug 19.
58. R Sennrich, & B Haddow, A Birch. *Neural machine translation of rare words with subword units*. arXiv preprint arXiv:1508.07909. 2015 Aug 31
59. MF McTear, Z Callejas, & D Griol. *The conversational interface*. Cham: Springer; 2016.
60. A Khanna, B Pandey, K Vashishta, K Kalia, B Pradeepkumar, & T Das. *A study of today's AI through chatbots and rediscovery of machine intelligence*. **International Journal of u-and e-Service, Science and Technology**. 2015 Jul 31;8(7):277-84.
61. SA Abdul-Kader, & Woods JC. *Survey on chatbot design techniques in speech conversation systems*. **International Journal of Advanced Computer Science and Applications**. 2015;6(7).
62. S Arsovski, H Osipyan, MI Oladele, & AD Cheok. *Automatic knowledge extraction of any Chatbot from conversation*. Expert Systems with Applications. 2019 Dec 15;137:343-8
63. Y Hahn, J Kim, S An, M Lee, & KS Choi. *Chatbot Who Wants to Learn the Knowledge: KB-Agent*. Semdeep/NLIWoD@ ISWC. 2018 Oct 9;4.
64. N Tavichaiyuth, & E Rattagan. *Developing Chatbots in Higher Education: A Case Study of Academic Program Chatbot in Thailand*.
65. YM Mohialden, MT Younis, & NM Hussien. *A Novel Approach to Arabic Chabot, Utilizing Google Colab and the Internet of Things: A Case Study at a Computer Center*. Webology. 2021 Dec;18(2).
66. Sousa FP. *The impact of social virtual presence agents and content-based product recommendation system on on-line customer purchase intention* (Doctoral dissertation), 2020

67. P Anki, A Bustamam, HS Al-Ash, & D Sarwinda. *Intelligent Chatbot Adapted from Question and Answer System Using RNN-LSTM Model*. **InJournal of Physics: Conference Series** 2021 Mar 1 (Vol. 1844, No. 1, p. 012001). IOP Publishing
68. HK Ahmed, & JA Hussein. *Design and Implementation of a Chatbot for Kurdish Language Speakers Using Chatfuel Platform*. **Kurdistan Journal of Applied Research**. 2020:117-35.
69. ZJ Tham. *A conversational question and answer chatbot for university* (Doctoral dissertation, UTAR).2022
70. L Zhou, J Gao, D Li, & HY Shum. *The design and implementation of xiaoice, an empathetic social chatbot*. *Computational Linguistics*. 2020 Mar 1;46(1):53-93.
71. M Nuruzzaman, & OK Hussain. *IntelliBot: A Dialogue-based chatbot for the insurance industry*. *Knowledge-Based Systems*. 2020 May 21;196:105810.
72. RK Behera, PK Bala, & A Ray. *Cognitive Chatbot for personalised contextual customer service: Behind the scene and beyond the hype*. *Information Systems Frontiers*. 2021 Jul 9:1-21.
73. EW Ngai, MC Lee, M Luo, PS Chan, & T Liang. *An intelligent knowledge-based chatbot for customer service*. *Electronic Commerce Research and Applications*. 2021 Nov 1;50:101098.
74. NP Patel, DR Parikh, DA Patel, & RR Patel. *AI and web-based human-like interactive university chatbot (UNIBOT)*. In2019 3rd international conference on electronics, communication and aerospace technology (ICECA) 2019 Jun 12 (pp. 148-150). IEEE.
75. U Arsenijevic, & M Jovic. *Artificial intelligence marketing: chatbots*. In2019 international conference on artificial intelligence: applications and innovations (IC-AIAI) 2019 (pp. 19-193). IEEE.
76. CC Ho, HL Lee, WK Lo, & KF Lui. *Developing a chatbot for college student programme advisement*. In2018 International Symposium on Educational Technology (ISET) 2018 Jul 31 (pp. 52-56). IEEE.
77. D Oklesinski, & W Jaworski. *Implementation of current state-of-the-art technology for obtaining a working Dialogue Agent*

78. N Shi, Q Zeng, & R Lee. *The design and implementation of Language Learning Chatbot with XAI using Ontology and Transfer Learning*. arXiv preprint arXiv:2009.13984. 2020 Sep 29.
79. R De Cicco, SC Silva, & FR Alparone. "It's on its way": *Chatbots applied for online food delivery services, social or task-oriented interaction style?*. **Journal of Foodservice Business Research**. 2021 Mar 4;24(2):140-64.
80. M Driss, I Almomani, L Alahmadi, L Alhajjam, R Alharbi, & S Alanazi. *COVIBOT: A Smart Chatbot for Assistance and E-Awareness during COVID-19 Pandemic*. arXiv preprint arXiv:2204.07851. 2022 Apr 16.
81. B Xu, & Z Zhuang. *Survey on psychotherapy chatbots. Concurrency and Computation: Practice and Experience*. 2022 Mar 25;34(7):e6170
82. S Bosse. *Distributed Serverless Chatbot Networks using Mobile Agents: A Distributed Data Base Model for Social Networking and Data Analytics*. In ICAART (1) 2021 (pp. 398-405).
83. AS Lokman, MA Ameen, & NA Ghani. *A conceptual IR chatbot framework with automated keywords-based vector representation generation*. In IOP Conference Series: Materials Science and Engineering, 2020 Feb 1 (Vol. 769, No. 1, p. 012020). IOP Publishing
84. A Khadija, FF Zahra, & Naceur A. *AI-Powered Health Chatbots: Toward a general architecture*. *Procedia Computer Science*. 2021 Jan 1;191:355-60.
85. N Shi, Q Zeng, & RS Lee. *The Design and Implementation of Intelligent English Learning Chatbot based on Transfer Learning Technology*. 2019
86. L Zhou, J Gao, D Li, & HY Shum. *The design and implementation of xiaoice, an empathetic social chatbot*. *Computational Linguistics*. 2020 Mar 1;46(1):53-93.
87. AR Asadi, & R Hemadi. *Design and implementation of a chatbot for e-commerce. Information Communication Technology and Doing Business*. 2018 Feb:1-0
88. PA Angga, WE Fachri, A Eleanita, & RD Agushinta. *Design of chatbot with 3D avatar, voice interface, and facial expression*. In 2015 international conference on science in information technology (ICSITech) 2015 Oct 27 (pp. 326-330). IEEE

89. D Kraus, V Reibenspiess, & A Eckhardt. *How voice can change customer satisfaction: a comparative analysis between e-commerce and voice commerce*. 14th International Conference on Wirtschaftsinformatik, February 24-27, 2019, Siegen, Germany
90. A Vegesna, P Jain, & D Porwal. *Ontology based chatbot (for e-commerce website)*. **International Journal of Computer Applications**. 2018 Jan;179(14):51-5.
91. W Nwankwo. *Interactive advising with bots: improving academic excellence in educational establishments*. **American Journal of Operations Management and Information Systems**. 2018 Mar 20;3(1):6-21.
92. V Subramanian, N Ramachandra, & N Dubash. *TutorBot: contextual learning guide for software engineers*. In 2019 IEEE/ACM 1st International Workshop on Bots in Software Engineering (BotSE) 2019 May 28 (pp. 16-17). IEEE.
93. A Palanica, P Flaschner, A Thommandram, M Li, & Y Fossat. *Physicians' perceptions of chatbots in health care: cross-sectional web-based survey*. **Journal of medical Internet research**. 2019 Apr 5;21(4):e12887.
94. T Anh. *Artificial Intelligence In e-Commerce: Case Amazon*. Thesis Centria University Of Applied Sciences Business Management May 2019
95. D Kaczorowska-Spychalska. *Chatbots in marketing*. Management Vol. 23, No. 1, 2019;23(1).
96. R. Khandale, S. Sombansi, S. Mishra, & M. F. Shaikh, and P. Mishra, "E-negotiator chatbot for e-commerce websites: implementation," **Journal of Applied Science and Computations**, vol. 6, no. 6, p. 489, 2019
97. S Angelov, & M Lazarova. *E-commerce distributed chatbot system*. In Proceedings of the 9th Balkan Conference on Informatics 2019 Sep 26 (pp. 1-8).
98. A Bhawiyuga, MA Fauzi, ES Pramukantoro, & W Yahya. *Design of e-commerce chat robot for automatically answering customer question*. In 2017 International Conference on Sustainable Information Engineering and Technology (SIET) 2017 Nov 24 (pp. 159-162). IEEE.
99. L Cui, F Huang S, Wei, C Tan, C Duan, & M Zhou. *Superagent: A customer service chatbot for e-commerce websites*. In Proceedings of ACL 2017, system demonstrations 2017 Jul (pp. 97-102).

- ^{100.} F Steinbauer, R Kern, & M Kröll. *Chatbots assisting German business management applications*. In International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems 2019 Jul 9 (pp. 717-729). Springer, Cham.
- ^{101.} H Joshi, V Agarwal, A Ghodke, D Gupta, & S Gaikwad. *Proposal of chat based automated system for online shopping*. **American Journal of Neural Networks and Applications**. 2017;3(1):1-4.
- ^{102.} A Nursetyo, & ER Subhiyakto. *Smart chatbot system for e-commerce assistance based on AIML*. In 2018 International Seminar on Research of Information Technology and Intelligent Systems (ISRITI) 2018 Nov 21 (pp. 641-645). IEEE.
- ^{103.} V Oguntosin, & A Olomo. *Development of an e-commerce chatbot for a university shopping mall*. Applied Computational Intelligence and Soft Computing. 2021 Mar 20;2021.

Do Not Copy, Lead City University, Nigeria

Chapter Three

Methodology

3.1 Research Approach

The methods involved in developing the chatbot for the customer support system, including the chatbot's architecture, components, and the data that passes through the chatbot system, will be detailed in detail in this chapter. Furthermore, the minimum hardware requirements and software needs will be covered.

3.2 Requirement Specification

Hardware Minimum Requirements: The minimum hardware requirements pertain to the physical features of the machine required to run the chatbot. The following are the features: at least 250 GB HDD, 4 GB RAM, and an Intel Pentium Dual-Core processor.

Software Requirements: These are the computer programmes and procedures needed to put the chatbot into action. The tools used include

- i. Spring boot: to implement the bot logic
- ii. Vaadin: to build a web UI to interact with the AIML bot
- iii. AIML(Artificial Intelligence Markup Language): was used to train the bot
- iv. detectLanguage java library was used for interpretation to any language of choice.

3.3 System Design

3.3.1. Chatbot Interface

Java is a platform-independent programming language. It means that we can run Java on the platforms that have a Java interpreter. It is the reason that makes the Java platform-independent.

The Java interpreter converts the Java bytecode (.class file) into the code understood by the operating system. Java interpreter is a computer program (system software) that implements the JVM. It is responsible for reading and executing the program. It is designed in such a way that it can read the source program and translate the source code instruction by instruction. It converts the high-level program into assembly language (machine language). AIML java interpreter library Program AB was used in this to design the interface

3.3.2. Web UI

A Web user interface or Web app allows the user to interact with content or software running on a remote server through a Web browser. The content or Web page is downloaded from the Web server and the user can interact with this content in a Web browser, which acts as a client. The distributed nature allows the content to be stored on a remote server, while the ubiquitous nature of the Web browser permits a convenient access to the content. The web UI was built using “vaadin” to interact with the AIML bot. Vaadin is an open-source web application platform for java. It supports two different programming models: server-side and client-side.

For this research, the web UI was built using vaadin to interact with the AIML bot. Vaadin is an open-source web application development platform for Java. Vaadin includes a set of Web Components, a Java web framework, and a set of tools that enable developers to implement modern web graphical user interfaces (GUI) using the Java programming language only (instead of HTML and JavaScript), TypeScript only, or a combination of both.

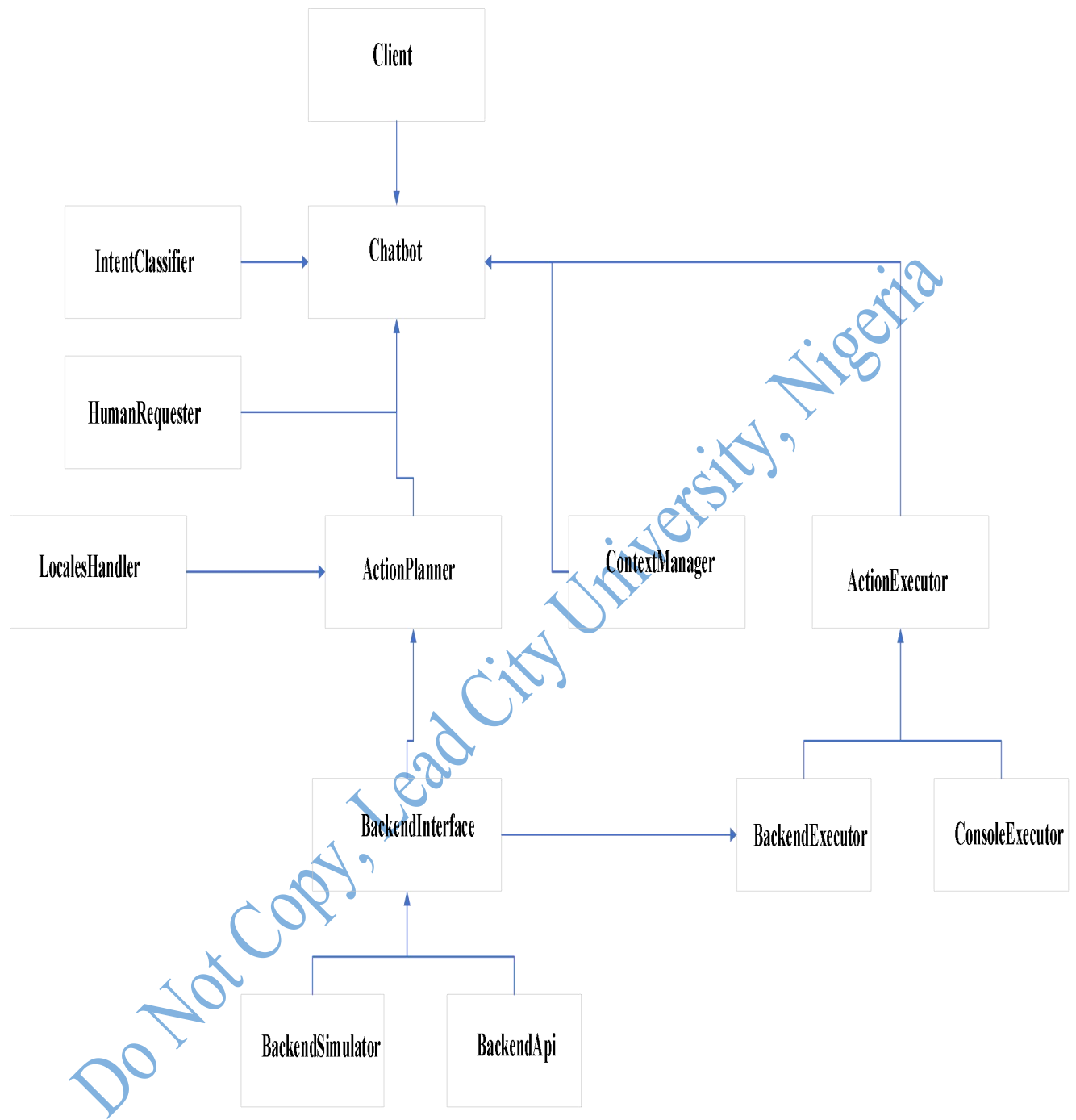


Figure 3.1 Chatbot Class Diagram

Source: Researcher Design 2023

3.3.3 Class diagram

This section will highlight and discuss the connections between all the agent's modules. The class diagram for the agent's modules is shown in Figure 3.1. As illustrated in this diagram, the chatbot is composed of several components, each of which addresses a distinct problem. At the top of the diagram, the component labelled Client is a generic client programme that makes use of the chatbot. It can be as simple as a terminal client or as complex as a dynamic online listener that retrieves new tickets issued by the system. The five primary components of the chatbot are the IntentClassifier, the HumanRequester, the ActionPlanner, the ContextManager, and the ActionExecutor. Additionally, it is demonstrated that the ActionPlanner has access to two submodules, one of which is shared with another submodule (BackendInterface)².

System Modules

This section will introduce the system's submodules by discussing their respective functions and providing information about their inputs and outputs.

Intent Classifier

This module is in charge of deducing the user intent from a message. It makes extensive use of neural networks in its normal operation. It takes a single user message as input and outputs a dictionary mapping each supported user intent to its associated probability.

Human Intervention Requester

This module is invoked immediately following the intent classifier. Its function is to request the intervention of a human customer service agent if the user's intent cannot be correctly identified. It is provided as a module to make switching between behaviours easier (For instance, one might want a more conservative chatbot when dealing with high profile customers that will request human intervention more often). It receives as input the entire context of a user dialogue as

provided by the context manager. It returns a boolean value that is true if the module determines that human intervention is required in the given context; otherwise, it returns false.

Context Manager

The module tasked with the responsibility of managing the structure that organises contexts. In this case, a context is defined as a collection of messages, as well as the metadata associated with them, that are useful for assisting a user with a single problem. This roughly corresponds to our intuitive concept of a single conversation/ticket in human terms.

Action Planner

The action planner's objective is to determine which actions the agent should take given a probability vector. It is responsible for monitoring the state of the backend and generating a payload containing the actions to be performed and in what order. In this case, the payload will be formatted as a JSON object². Additionally, it oversees constructing the actual response that will be sent to the user. It makes use of two submodules: one for handling locales and another for the backend interface

3.3.4 Development of the Chatbot with AIML

The main aim of the AIML language is to simplify the modelling process of dialogue. It gives access to stimulus response. In addition to that, AIML is an XML-based markup language. AIML works by defining a class object that is responsible for modelling the patterns of conversation. It is the most-used chatbot language due to its simplicity, ease of learning, ease of implementation and the availability of pre-authored AIML sets. It is a simple word pattern-matcher that generates a response based on a query. The AIML robot responds according to connection between the questions set by the user and knowledge located in AIML files.

3.4 Research Method

3.4.1 Data Collection

The messages dataset was extracted from already made AIML file for FAQ questions. Artificial Intelligence Markup Language, or AIML, is an XML-based language used to program natural language software agents, such as chatbots. The already made questions are shown below using bank account, Nigeria Immigration Service, FIRS and custom duty question:

Bank Account “Can I Open An Account While Outside The Country?”, “Is It Possible To Open An Account In The Country And Operate The Account While Out Of The Country?”, “How Can I Open A Salary Account?”, “How Long Does It Take Before My Account Becomes Inactive?”, Can A Savings Account Become Dormant?, How Long Does It Take To Open An Account In Banks?, What Are The Requirements For A Change Of Address in Banks?.

Nigerian Immigration Services “Will I have to start the process for my passport or visa in the bank?”, Will I get my Visa or Passport once payment is made in the bank?

Federal Inland Revenue Service Payments “What is TIN?”, Can payment be made in any Bank's branch? “What documents required for payment?” “How can a new taxpayer get the TIN? Must a taxpayer be assessed before payment?”

Customs Duty Payments “Can payment be made at any bank?, What are the documentation required for payment? What is SGD Assessment Notice?”, “Who issued the SGD Assessment Notice?”

3.4.2 Preprocessing

The following are the preprocessing steps in the order in which they will be performed:

- i. Reduce the case of all characters.

- i. Remove superfluous punctuation characters (e.g. "!,^ ...)
- ii. Tokenize the texts by using words as tokens.. A word is defined in this context as a contiguous sequence of characters enclosed in white space.
- iii. Compute the vocabulary of the texts (i.e. the set of tokens used).
- iv. Replace each token in the sequences with its vocabulary index.
- v. Reduce the length of the sequences to a fixed value of 100 tokens.

3.4.3 Training

AIML(Artificial Intelligence Markup Language) was used to create, train, and monitor the conversational progress of the bot. Furthermore, it has a webhook feature that will let go bot go live and be present on the chatbot user interface. Training the bot can be referred to as giving it a brain to understand its creation purpose. The foundation of the bot's brain can be derived from a term called intents¹. An intent is a box of sentences or expressions that all have the same meaning. When a user sends a message to bot, the algorithm embedded within the API compares the user's input to the expressions housed within the bot's intents. It picks the intent with the highest similarity rate to the user's input and then selects a suitable reply for the bot to give to the user. The reply to various intents is embedded within the intents. To further expand the knowledge base of the bot, the API has a feature that accommodates external data. This external data is in the customer support datasets, which can be accessed from the admin login portal and DBMS. The bot undergoes a lot of supervised learning during its training process, where monitoring its conversational progress finds expression.

3.4.4 Implementation of Software

Java was used as the primary programming language for this research in order to implement the chatbot. Java is an easy-to-learn language that was deemed appropriate for this situation. The

scikit-learn library is a minor (relative to this project) but nonetheless significant library². It is used in this context for a variety of utilities, including segmenting a dataset into train validation and test sets.

3.4.5 Generating Responses

Responses was generated using a retrieval-based approach. Indeed, this enables the agent's maintainers to ensure that the agent does not make an offensive statement to the user. The Action Planner module is responsible for retrieving responses, and in this case, it contains all the logic that is triggered when the user intent is correctly identified. For instance, it may contact the backend to verify that the user's account is not blacklisted or that the user's most recent withdrawal was delayed since it was made to an international bank account. The responses are structured similarly to how customer support agents might respond in real life. Thus, they are formatted to resemble formal e-mails, as this is the primary source of tickets received by customer service.

3.4.6 Agent Environment

The agent environment is composed of several distinct components. Among them are user messages, which are a type of dynamic input that the agent may receive at any time. They are composed of a string containing the user's actual text and a metadata structure containing additional information such as a pointer connecting the message to the structure representing the conversation to which it belongs, as well as the time the message was sent, the platform on which it was sent, and so on. The chatbot can only read the data it contains and has no way of altering it. Additionally, the agent has access to the company's backend, which contains additional information about the user and the state of the database. The agent has the ability to both inspect and modify certain aspects of the backend. Also, the chatbot can respond to users to

obtain additional information or simply to inform them that their request has been accepted and handled appropriately. Figure 3.2 depicts the agent's environment visually.

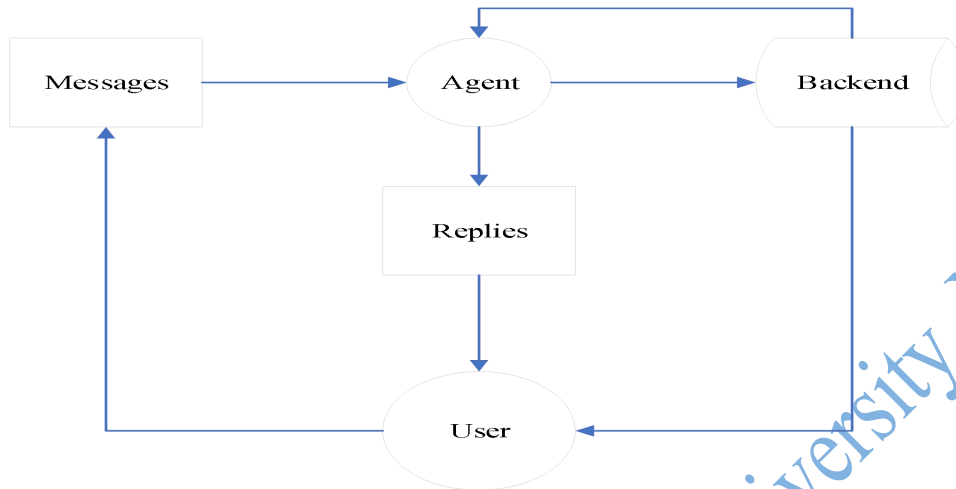


Figure 3.2: Agent Environment Model

Source: Researcher Design 2023

3.5 Performance Evaluation of the Developed System

The developed system was evaluated using the manual software testing technique by a test engineer⁴. Manual testing is a type of software testing in which test cases are run by manually rather than using an automated tool. Manual testing is done in a variety of ways, including white box, black box, and gray box testing. For the purpose of this research, the non-functional Black Box Testing was used. Black box testing is a type of software testing that looks at the software's functionality without looking at its internal structure or code. In this approach, the tester chooses a function and inputs a value to verify its functioning, then verifies whether the function produces the anticipated result. Table 3.1 shows test scenarios for the developed system evaluation. Hence the following test were done to evaluate the performance of the system.

- i. Performance and Scalability Testing (Response Time, Stability, Load)

ii. Usability Testing (Easy to understand, Easy to access, Faster to Access, Effective Navigation)

iii. Compatibility and Reliability Testing (Software, Hardware. Network, Mobile)

Table 3.1: Test Scenarios for the Developed Chatbot

S/N	Test Type	Score (%)
1.	Performance	<ul style="list-style-type: none">i. Response Timeii. Stabilityiii. Loadiv. Reliability
2.	Usability	<ul style="list-style-type: none">i. Easy to Understandii. Easy to Accessiii. Faster to Accessiv. Effective Navigation
3.	Compatibility	<ul style="list-style-type: none">i. Softwareii. Hardwareiii. Networkiv. Mobile
4.	Scalability	<ul style="list-style-type: none">i. Throughputii. Memory Usageiii. CPU Usageiv. Network Usage

Endnotes

¹ V Oguntosin, & A Olomo. *Development of an e-commerce chatbot for a university shopping mall*. Applied Computational Intelligence and Soft Computing. 2021 Mar 20;2021.

² F Peters. Master thesis: *Design and implementation of a chatbot in the context of customer support*. Masters Dissertation.2018.

³ J VanderPlas. *Python data science handbook: Essential tools for working with data*. " O'Reilly Media, Inc."; 2016 Nov 21.

⁴ www.javatpoint.com/compatibility-testing

Do Not Copy, Lead City University, Nigeria

Chapter Four

Results and Discussion of Findings

4.1 Result of Chatbot Design

In this chapter, the results from the implementations of our designs are presented. For clarity, the results are presented based on the objectives (including supporting ones) outlined at the beginning of this study.

4.1.1 Result on Java Interpreter

The chatbot was developed using a Java interpreter. It is a computer program that converts high-level program statement into Assembly Level Language¹. It is designed to read the input source program and then translate the source program instruction by instruction. AIML java interpreter library Program AB was added which main function is to match input and output predefined in the AIML file¹. Below is the snippet code for the Java interpreter used. The full programming code is attached in Appendix I.

```
<category>
<pattern>CAN I OPEN AN ACCOUNT WHILE OUTSIDE THE COUNTRY</pattern>
<template>You can open an account online here while out of the country if you already
have a Bank Verification Number (BVN). If you do not have a BVN, you will still receive
an account number, but the account will remain restricted until you complete the
required documentation.</template>
</category>
```

This would lead to the following dialogue;

User: *Can I open an account while outside the country?*

Bot : *You can open an account online here while out of the country if you already have a bank*

verification number (BVN) . if you do not have a BVN, you will still receive an account number, but the account will remain restricted until you complete the required documentation.

4.1.2 Result of Spring Boot Project

The spring boot project was created using Spring initializr and implementing the Bot logic. Spring Initializr is a web-based tool used to generate the structure of the Spring Boot Project². It offers extensible API for creating JVM-based projects. It also provides various options for the project that are expressed in a metadata model. The metadata model allows the configuration of the list of dependencies supported by JVM and platform versions, etc. It serves its metadata in a well-known that provides necessary assistance to third-party clients. Spring Initializr has the following module; initializr-actuator which provides additional information and statistics on project generation. It is an optional module, initializr-bom used to control the versions of a project's dependencies². It provides a central place to define and update those versions. It provides flexibility to add a dependency in our module without worrying about the versions, initializr-docs: It provides documentation, initializr-generator: It is a core project generation library, initializr-generator-test provides a test infrastructure for project generation, initializr-metadata provides metadata infrastructure for various aspects of the projects, initializr-service-example provides custom instances, initializr-version-resolver an optional module to extract version numbers from an arbitrary POM, initializr-web provides web endpoints for third party clients. Below is the snippet code for the Java interpreter used². The full programming code is attached in Appendix I.

```
import org.springframework.boot.builder.SpringApplicationBuilder;  
import org.springframework.boot.web.servlet.support.SpringBootServletInitializer;
```

```
public class ServletInitializer extends SpringBootServletInitializer {
```

```
@Override
protected SpringApplicationBuilder configure(SpringApplicationBuilder application) {
    return application.sources(Application.class);
}
```

4.1.3 Result on AIML file for FAQ questions

An AIML file contains Artificial Intelligence Markup Language (AIML), which is used to define the personality of a chat robot, or chatbot³. It stores AIML, which is saved in XML format and is used to specify the heuristic conversation rules for a chatbot. AIML files can be created and edited by any plain text editor. A chatbot typically has a collection of AIML files that guide its conversation pattern and give the robot "character. Below are some FAQ used in designing the chatbot. Below is the snippet code for the Java interpreter used. The full programming code is attached in Appendix I.

```
category>
  <pattern>HOW LONG DOES IT TAKE BEFORE MY ACCOUNT BECOMES
  INACTIVE</pattern>
  <template>Current accounts become dormant after 6 months of inactivity.</template>
</category>
```

Alice

CAN I OPEN AN ACCOUNT WHILE OUTSIDE THE COUNTRY?



You



You can open an account online here while out of the country if you already have a Bank Verification Number (BVN). If you do not have a BVN, you will still receive an account number, but the account will remain restricted until you complete the required documentation.

Alice

(a)

IS IT POSSIBLE TO OPEN AN ACCOUNT IN THE COUNTRY AND OPERATE THE ACCOUNT WHILE OUT OF THE COUNTRY?



You



Yes. The account can be operated through bank card products or internet banking solution.

Alice

(b)

HOW CAN I OPEN A SALARY ACCOUNT?

You

Alice

Your company / organisation will write to bank to introduce the prospective account holder to the bank for account opening.

(c)

HOW LONG DOES IT TAKE TO OPEN AN ACCOUNT IN BANKS?

You

Alice

Accounts are opened within 24 hours upon submission of complete documentation.

(d)

HOW LONG DOES IT TAKE TO OPEN AN ACCOUNT IN BANKS?

You

Alice

Accounts are opened within 24 hours upon submission of complete documentation.

(e)

WHAT DOCUMENTS REQUIRED FOR PAYMENT?

You

Alice

Yes, you can pay your tax at any Bank's branch.

(f)

Alice: Customer TIN, Assessment Form, WHT schedule.

You: HOW CAN A NEW TAXPAYER GET THE TIN?

(g)

Alice: Customer TIN, Assessment Form, WHT schedule.

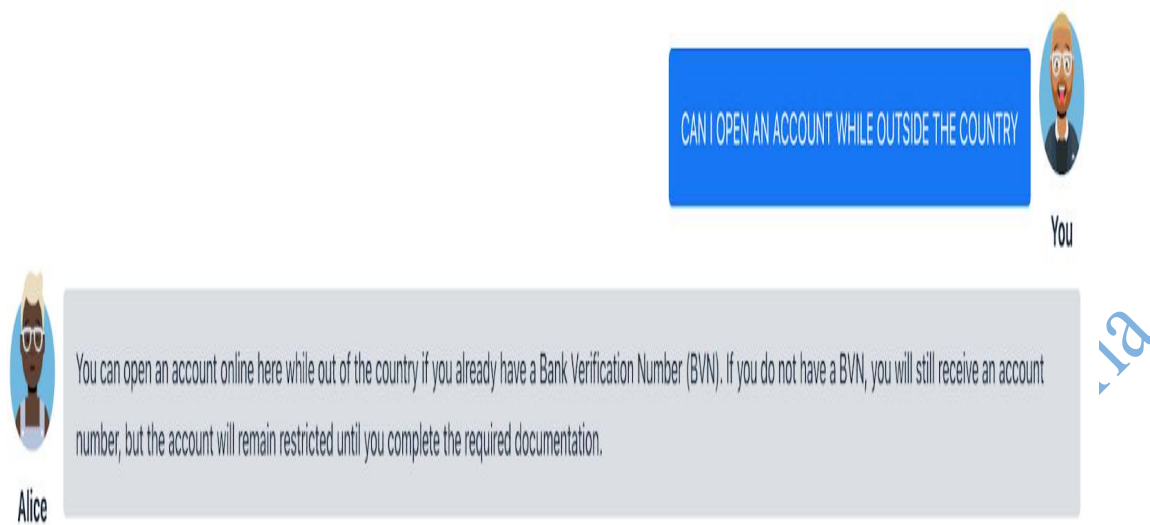
You: HOW CAN A NEW TAXPAYER GET THE TIN?

(h)

Alice: Taxpayers can get their Tax Identification Number (TIN) from any Federal Inland Revenue Service (FIRS) tax office.

You: MUST A TAXPAYER BE ASSESSED BEFORE PAYMENT?

(i)



(j)

Figure 4.1 (a-j): Snapshot of the Result of FAQ questions and Chatbot Responses

Source: Researcher Design 2023

4.1.4 Result on Translation from any Language to English

Google script was written to translate from any language to English for the bot to understand and send the response in the preferred language of the user. For this research, the Bot was tested to translate to Yoruba language as shown in figure 4.2. Below is the snippet code for the google script. The full code is attached in Appendix I

```
// Post requests are processed within the doPost(e) method in Google Apps Script
function doPost(e) {

  // In POST requests, parameters and variables are contained in e.postData.contents as a default
  var datas = JSON.parse(e.postData.contents);

  // This is the JSON object that will return as the response we created.
  var response = {
    data: [],
    message: ""
  };
}
```



Figure 4.2: Snapshot of the Result of Response in the Preferred Language of Use

Source: Researcher Design 2023

4.1.5 Result of Performance Evaluation of the Developed System

Table 4.2: Performance Evaluation Table⁴.

S/N	Test Type		Score (%)
1	Performance	i. Response Time	90
		ii. Stability	80
		iii. Load	70
		iv. Reliability	90
2	Usability	i. Easy to Understand	95
		ii. Easy to Access	90
		iii. Faster to Access	90
		iv. Effective Navigation	85
3	Compatibility	i. Software	95
		ii. Hardware	90
		iii. Network	80
		iv. Mobile	90
4	Scalability	i. Throughput	80
		ii. Memory Usage	1.8
		iii. CPU Usage	0
		iv. Network Usage	0

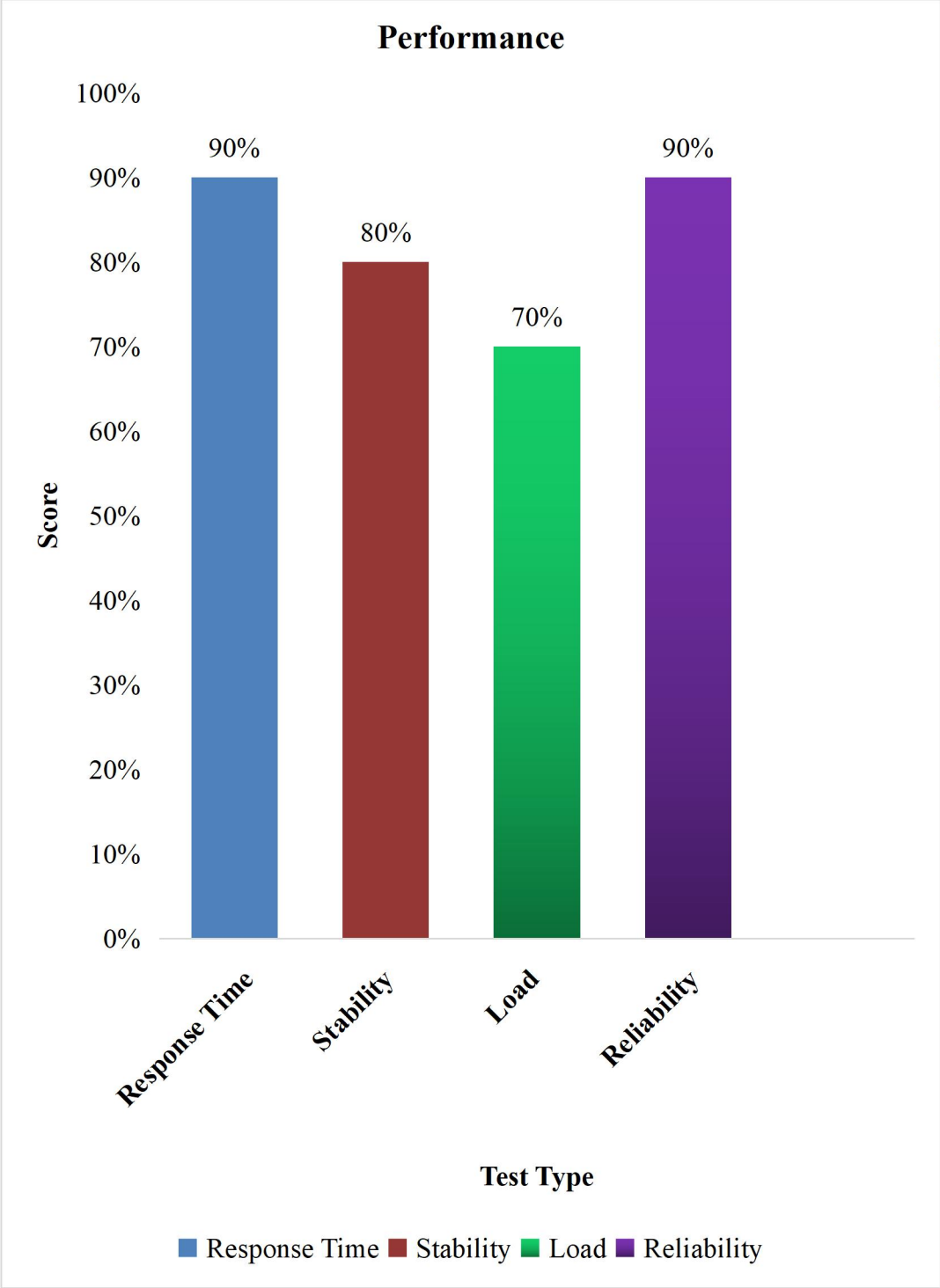


Figure 4.3: Performance Testing Chart of Developed Chatbot

Source: Researcher Design 2023

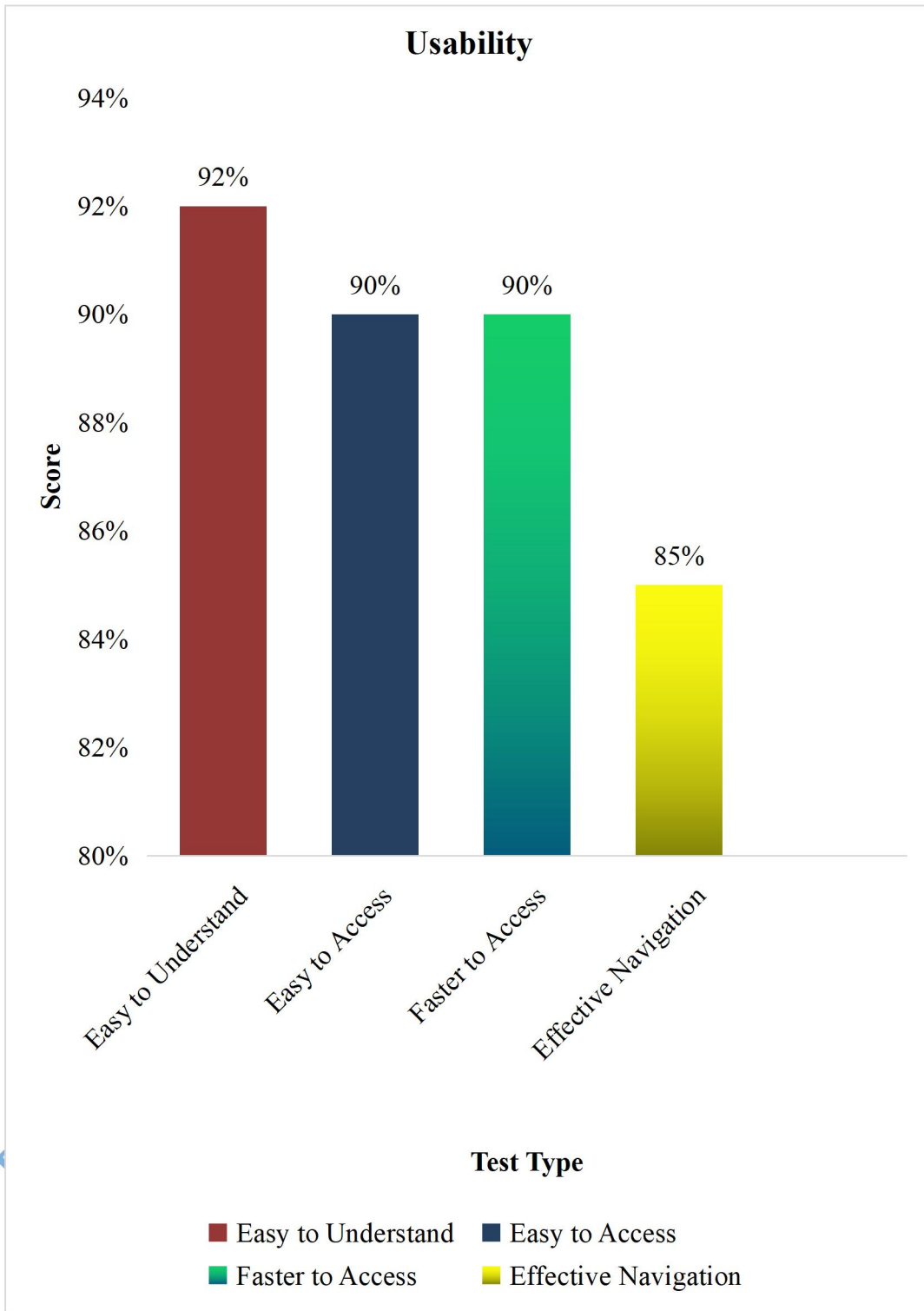


Figure 4.4: Usability Testing Chart of Developed Chatbot

Source: Researcher Design 2023

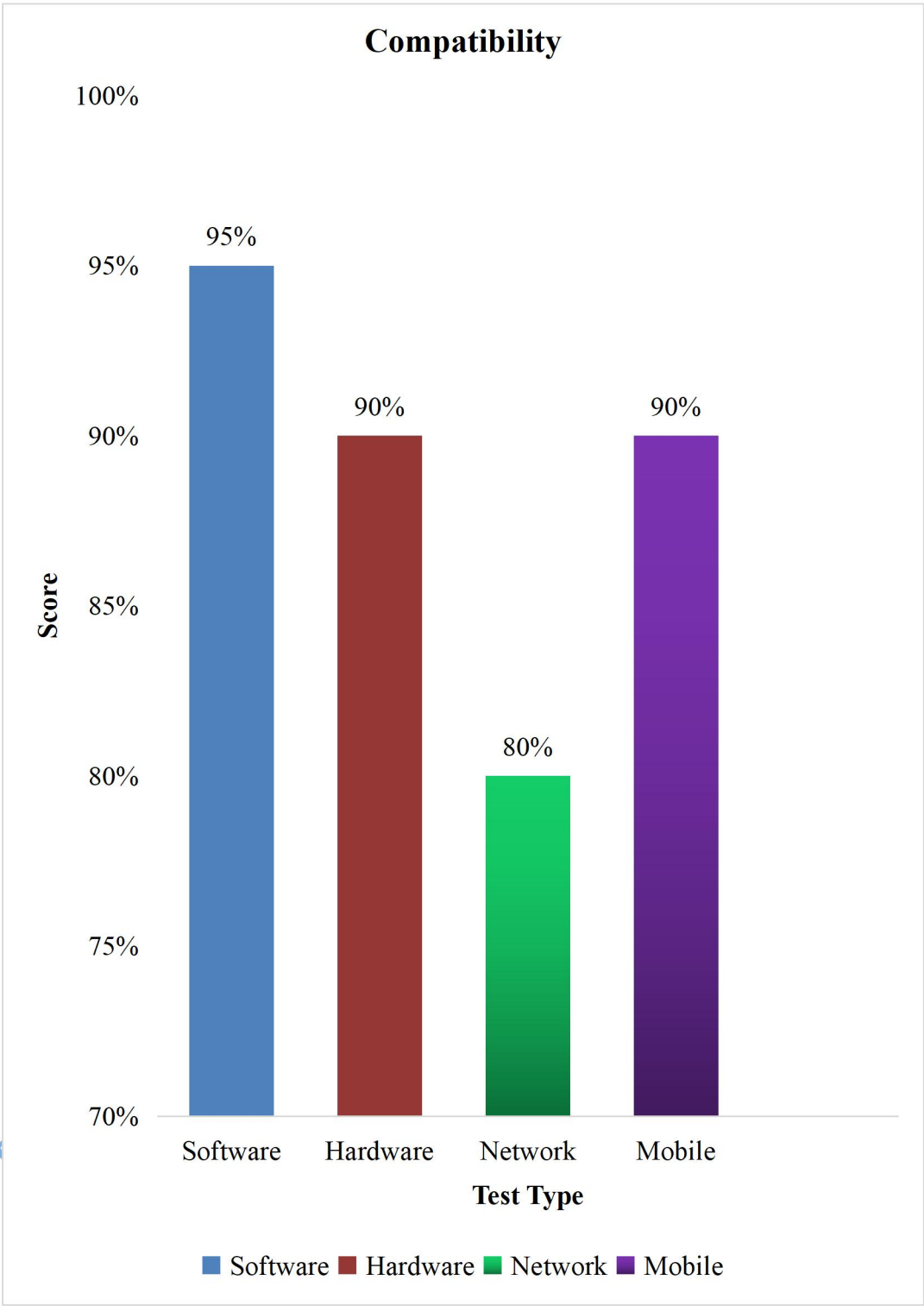


Figure 4.5: Compatibility Test Chart of the Developed Chatbot

Source: Researcher Design 2023

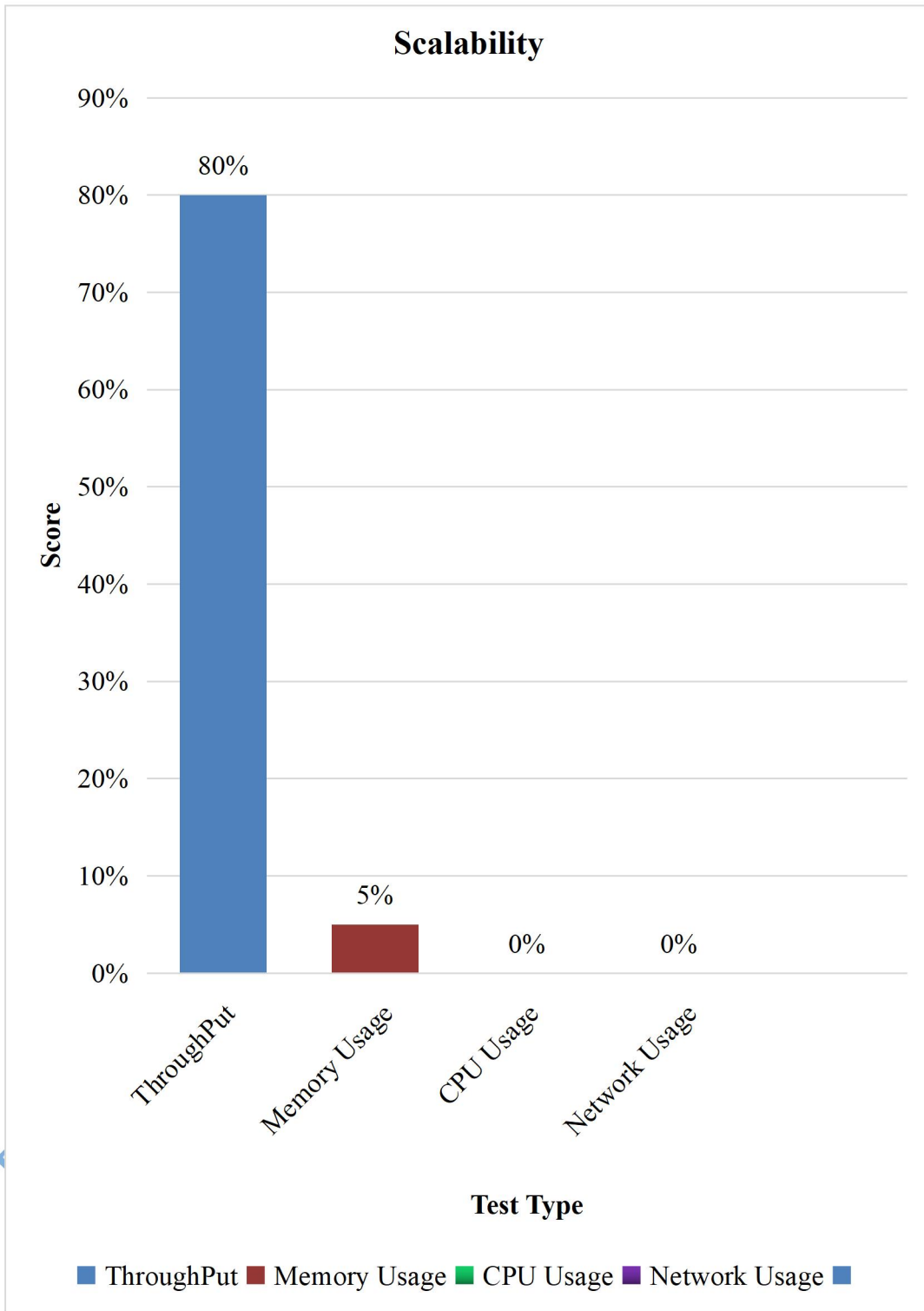


Figure 4.6: Scalability Test Chart of Developed Chatbot

Source: Researcher Design 2023

4.2 Discussions of Findings

The chatbot was developed using AIML java interpreter library Program AB which helps match input and output predefined in the AIML file. This would lead to the following dialogue shown in Figure 4.1 (a-j). The spring boot project was created using Spring initializr and implementing the Bot logic. AIML (Artificial Intelligence Markup Language) was used to preprocess and train the bot using ready-made AIML file for FAQ questions. Also, vaadin was used to build a web UI to interact with the trained AIML bot. Finally, a google script was written to translate from any language to English for the bot to understand and send the response in the preferred language of the user. The speed of the bot network however depends on the network used. Any failure in the network may delay the response time of the system. Figure 4.3 to 4.5 shows the test chart to evaluate the developed chatbot. The chart is a plot of test score against the test type. The evaluation was done by a test engineer to ascertain the performance of the design.

4.2.1 Performance Testing

Response time: It is the time taken by the server to respond to the request. The response time of the bot is dependent of the network used for internet.

Load testing is used to check the performance of an application by applying some load which is either less than or equal to for the stability factor. The prototype design was tested with 10 users at the same time on different web page. The goal is to respond immediately with the numbers of users. However, it showed the capacity to be stable with a score of 70%.

Stability is checking the performance of the design by applying the load for a particular duration of time. The design was stable for the load of 10 users at the response time of 2-4 second. However, as the load was increased gradually up to 20 users to check the maximum load with the

target response time, the response delayed for about 2-5 seconds. This gave a stability of 80% for 10-15 users at a go. Load testing more than 20 users was not used

Reliability is described as the probability of defect-free operation of the prototype design for a particular time in a specific environment. The design was reliable at a good network speed with attenuation. Hence a score of 90%

4.2.2 Usability Testing

This is the checking the user-friendliness, efficiency, and accuracy of the design. The design is easy to understand, it requires little or no basic training and can be used even if one is not a professional. Easy to access since the component required are not expensive and mostly open source, faster to access i.e. the design is faster while accessing, which means that the response time is quick and effective navigation, i.e. good internal linking and informative header and footer. Hence having a score of 95%, 90%, 90%, 85% respectively.

4.2.3 Compatibility Testing

Software: The design is compatible with different operating systems, both for forward compatibility and backward compatibility having a score of 95%. Also, with different browsers like Google Chrome, Firefox, and Internet Explorer.

Win 7 → Win 8 → Win 8.1 → Win 10 (Forward Compatibility)

Window XP → Vista → Win 7 → Win 8 → Win 8.1 (Backward Compatibility).

Hardware: The design is compatible with hardware of different sizes such as RAM, hard disk, processor, and the graphic card with a score of 90%

Mobile: The design is compatible with mobile platforms such as iOS and Android. The mobile app for the design does not run on all Android phones less than Android 9, runs efficiently on iOS and all global phones. Thus, having a score of 90%

Network: Compatibility with different network parameters such as operating speed, bandwidth, and capacity with a score of 80%.

4.2.4 Scalability Testing

Throughput: The design has a Throughput of 80%, because it can respond in a second effectively and efficiently.

Do Not Copy, Lead City University, Nigeria

Endnotes

¹- Al-Ghadhban D, & N Al-Twairsh. *Nabiha: an Arabic dialect chatbot*. **International Journal of Advanced Computer Science and Applications**. 2020;11(3).

²- F Gutierrez. *Spring Boot*. In *Spring Cloud Data Flow 2021* (pp. 9-31). Apress, Berkeley, CA.

³- BR Ranoliya, N Raghuwanshi, & S Singh. *Chatbot for university related FAQs*. In 2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI) 2017 Sep 13 (pp. 1525-1530). IEEE.

⁴- R A Badru, A A Waheed, O A Akinmoluwa, & O R Obayemi. *Generation of Surveillance Networked Nodes for Oil Pipelines' Theft* **International Journal of Recent Engineering Science**, 8(5), 21-26.

Do Not Copy, Lead City University, Nigeria

Chapter Five

Conclusion

5.1 Summary of Findings

The central aim of this thesis was to develop an automated customer support system (chatbot) in an IT driven organization to solve web-based financial and payment application services. For this purpose, a literature review was conducted, and the findings were translated to the analysis afterwards. This examination enabled the author to gain knowledge about the usefulness and appropriateness of chatbot applications nowadays and to provide a recommendation for organisations operating in the service industry. Although chatbots have already been in use for some decades, it could be said that most of them still face some major problems concerning day-to-day business. One issue includes those users are always heterogeneous which implies that they are individuals and need answers to be personalised. The chatbots used in this research have been set up for solving the simplest queries and forward customers which have a specific enquiry, and thereby minimising labour resources in the long run. The system is made up of a spring boot to implement the bot logic; Vaadin to build a web UI to interact with the AIML bot, AIML(Artificial Intelligence Markup Language) used to train the bot and detectLanguage java library which was used for interpretation to any language of choice.

5.2 Conclusion

The bot was able to answer customer questions, enquiries and complaints and the response time of the bot depends on the strength of the network since it is web based. The system had a good response time, scalability, and reliability. Hence, the system provided a simple, cheaper and durable customer financial and payment application service.

5.3. Recommendations

Based on the findings from this study, the following recommendations were made:

- i. It is recommended that routine queries be standardised and hence deliver a high level of performance. Chatbots will not be able to solve all queries, especially in the financial and payment application services.
- ii. It is recommended that a company incorporating a chatbot should make sure that the chatbot is highly secure since chatbots have only been used commercially for the last few years. Consequently, there have already been issues with cyber-attacks.

5.4 Contribution to Knowledge

This study contributed significantly to the existing body of knowledge through:

- i. Assisting in the development of customer support applications that reduce customer complaints and wait times in financial and payment application services.

5.5 Suggestions for Further Research

The following are the suggestions for further research:

- i. Additional languages could be trained to cover a wider user base
- ii. A different dynamic version of the chatbot could be designed. This variant would need to be more reactive by displaying more small-talk capabilities and creating more direct and customised interactions with customers

- iii. In case the user is unable to clarify itself, instead of directly requesting a human operator, the chatbot could suggest potential problems to the user in the form of a selection list with common problems. There could also be the option to request a human agent in this list to prevent users from getting frustrated with the bot.

Do Not Copy, Lead City University, Nigeria

Bibliography

Book

Barfield W, Pagallo U, & editors. Research handbook on the law of artificial intelligence. Edward Elgar Publishing; 2018 Dec 28.

Gutierrez F. Spring Boot. In Spring Cloud Data Flow 2021 (pp. 9-31). Apress, Berkeley, CA

Natale S. Deceitful media: Artificial intelligence and social life after the Turing test. Oxford University Press, USA; 2021 Mar 17.

International Conference

Adamopoulou E, & Moussiades L. *An overview of chatbot technology*. In IFIP International Conference on Artificial Intelligence Applications and Innovations 2020 Jun 5 (pp. 373-383). Springer, Cham.

Ahirwar GK. Chatterbot: technologies, tools and applications. In High Performance Vision Intelligence 2020 (pp. 203-213). Springer, Singapore.

Ahmed HK, & Hussein JA. Design and Implementation of a Chatbot for Kurdish Language Speakers Using Chatfuel Platform. Kurdistan Journal of Applied Research. 2020:117-35.

Allal-Chérif O, Aranega AY, & Sánchez RC. Intelligent recruitment: How to identify, select, and retain talents from around the world using artificial intelligence. Technological Forecasting and Social Change. 2021 Aug 1;169:120822.

Al-Madi, N.A., Maria, K.A., Al-Madi, M.A., Alia, & M.A. and Maria, E.A., 2021, July. An intelligent Arabic chatbot system proposed framework. In *2021 International Conference on Information Technology (ICIT)* (pp. 592-597). IEEE

Angelov S, & Lazarova M. E-commerce distributed chatbot system. In Proceedings of the 9th Balkan Conference on Informatics 2019 Sep 26 (pp. 1-8).

Angga PA, Fachri WE, Eleanita A, & Agushinta RD. Design of chatbot with 3D avatar, voice interface, and facial expression. In 2015 international conference on science in information technology (ICSITech) 2015 Oct 27 (pp. 326-330). IEEE

Anh T. Artificial intelligence in e-commerce: Case Amazon. Thesis CENTRIA UNIVERSITY OF APPLIED SCIENCES. Business Management, May 2019

Anita Nathania P, & Yulia VY. Android Based Chatbot and Mobile Application for Tour and Travel Company. IJCT. 2018:21

Anki P, Bustamam A, Al-Ash HS, & Sarwinda D. Intelligent Chatbot Adapted from Question and Answer System Using RNN-LSTM Model. In **Journal of Physics: Conference Series** 2021 Mar 1 (Vol. 1844, No. 1, p. 012001). IOP Publishing.

- Arsenijevic U, & Jovic M. Artificial intelligence marketing: chatbots. In 2019 international conference on artificial intelligence: applications and innovations (IC-AIAI) 2019 (pp. 19-193). IEEE.
- Arsovski S, Osipyan H, Oladele MI, & Cheok AD. Automatic knowledge extraction of any Chatbot from conversation. *Expert Systems with Applications*. 2019 Dec 15;137:343-8
- Asadi AR, & Hemadi R. Design and implementation of a chatbot for e-commerce. *Information Communication Technology and Doing Business*. 2018 Feb:1-0.
- Badru RA, Waheed A A, Akinmoluwa OA, & Obayemi OR. Generation of Surveillance Networked Nodes for Oil Pipelines' Theft *International Journal of Recent Engineering Science*, 8(5), 21-26.
- Barber J, Yu X, Zamore LK, Lin J, Jazayeri V, Erlich S, Savor T, & Stumm M. Bladerunner: Stream Processing at Scale for a Live View of Backend Data Mutations at the Edge. In *Proceedings of the ACM SIGOPS 28th Symposium on Operating Systems Principles* 2021 Oct 26 (pp. 708-723).
- Behera RK, Bala PK, & Ray A. Cognitive Chatbot for personalised contextual customer service: Behind the scene and beyond the hype. *Information Systems Frontiers*. 2021 Jul 9:1-21.
- Bhawiyuga A, Fauzi MA, Pramukantoro ES, & Yahya W. Design of e-commerce chat robot for automatically answering customer question. In *2017 International Conference on Sustainable Information Engineering and Technology (SIET) 2017 Nov 24* (pp. 159-162). IEEE.
- Bosse S. Distributed Serverless Chatbot Networks using Mobile Agents: A Distributed Data Base Model for Social Networking and Data Analytics. In *ICAART (1) 2021* (pp. 398-405).
- Chrysovelidis G. Designing a chatbot for tourism. Thesis, International Hellenic University. 2020
- Cui L, Huang S, Wei F, Tan C, Duan C, & Zhou M. Superagent: A customer service chatbot for e-commerce websites. In *Proceedings of ACL 2017, system demonstrations 2017 Jul* (pp. 97-102).
- Cunningham-Nelson S, Boles W, Trouton L, & Margerison E. A review of chatbots in education: practical steps forward. In *30th Annual Conference for the Australasian Association for Engineering Education (AAEE 2019): Educators Becoming Agents of Change: Innovate, Integrate, Motivate 2019* (pp. 299-306). Engineers Australia.
- D'Haro LF, Kim S, Yeo KH, Jiang R, Niculescu AI, Banchs RE, & Li H. CLARA: a multifunctional virtual agent for conference support and touristic information. In *Natural language dialog systems and intelligent assistants 2015* (pp. 233-239). Springer, Cham.

- De Cicco R, Silva SC, & Alparone FR. "It's on its way": Chatbots applied for online food delivery services, social or task-oriented interaction style?. *Journal of Foodservice Business Research*. 2021 Mar 4;24(2):140-64.
- De Gennaro M, Krumhuber EG, & Lucas G. Effectiveness of an empathic chatbot in combating adverse effects of social exclusion on mood. *Frontiers in psychology*. 2020 Jan 23;10:3061.
- Driss M, Almomani I, Alahmadi L, Alhajjam L, Alharbi R, & Alanazi S. COVIBOT: A Smart Chatbot for Assistance and E-Awareness during COVID-19 Pandemic. arXiv preprint arXiv:2204.07851. 2022 Apr 16.
- Español SJ, Fronda Jr R, Villarica MR, & Intal GL. SHOP BUDDY Enhanced Automated Chatbot for Any E-commerce Website with Data Analytics Functionalities.
- Fernández-Avelino J, Alor-Hernández G, Paredes-Valverde MA, & Sánchez-Morales LN. Developing Chatbots for Supporting Health Self-Management. In *Handbook of Research on Natural Language Processing and Smart Service Systems 2021* (pp. 135-156). IGI Global.
- Følstad A, & Skjuve M. Chatbots for customer service: user experience and motivation. In *Proceedings of the 1st international conference on conversational user interfaces 2019 Aug 22* (pp. 1-9)
- Ghimire A, Thapa S, Jha AK, Adhikari S, & Kumar A. Accelerating business growth with big data and artificial intelligence. In *2020 Fourth International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC) 2020 Oct 7* (pp. 441-448). IEEE.
- Gutierrez F. *Spring Boot*. In *Spring Cloud Data Flow 2021* (pp. 9-31). Apress, Berkeley, CA.
- Hahm Y, Kim J, An S, Lee M, & Choi KS. Chatbot Who Wants to Learn the Knowledge: KB-Agent. *Semdeep/NLIWoD@ ISWC*. 2018 Oct 9;4
- Hayward KJ, & Maas MM. Artificial intelligence and crime: A primer for criminologists. *Crime, Media, Culture*. 2021 Aug;17(2):209-33.
- Hill J, Ford WR, & Farreras IG. Real conversations with artificial intelligence: A comparison between human-human online conversations and human-chatbot conversations. *Computers in human behavior*. 2015 Aug 1;49:245-50.
- Ho CC, Lee HL, Lo WK, & Lui KF. Developing a chatbot for college student programme advisement. In *2018 International Symposium on Educational Technology (ISET) 2018 Jul 31* (pp. 52-56). IEEE.
- Hsu I, & Yu JD. A medical Chatbot using machine learning and natural language understanding. *Multimedia Tools and Applications*. 2022 Mar 18:1-23
- Huang MH, & Rust RT. Engaged to a robot? The role of AI in service. *Journal of Service Research*. 2021 Feb;24(1):30-41

Huang X, & CIS A. Chatbot: Design, Architecture, And Applications

Hussain S, Ameri Sianaki O, & Ababneh N. A survey on conversational agents/chatbots classification and design techniques. In Workshops of the International Conference on Advanced Information Networking and Applications 2019 Mar 27 (pp. 946-956). Springer, Cham.

Ifinedo P, Pyke J, & Anwar A. Business undergraduates' perceived use outcomes of Moodle in a blended learning environment: The roles of usability factors and external support. *Telematics and Informatics*. 2018 Apr 1;35(1):93-102

Joshi H, Agarwal V, Ghodke A, Gupta D, & Gaikwad S. Proposal of chat based automated system for online shopping. *American Journal of Neural Networks and Applications*. 2017;3(1):1-4.

Lokman AS, Ameen MA, & Ghani NA. A conceptual IR chatbot framework with automated keywords-based vector representation generation. In *IOP Conference Series: Materials Science and Engineering 2020 Feb 1* (Vol. 769, No. 1, p. 012020). IOP Publishing.

Maroengsit W, Piyakulpinyo T, Phonyiam K, Pongnumkul S, Chaovalit P, & Theeramunkong T. A survey on evaluation methods for chatbots. In *Proceedings of the 2019 7th International Conference on Information and Education Technology 2019 Mar 29* (pp. 111-119)

Patel NP, Parikh DR, Patel DA, & Patel RR. AI and web-based human-like interactive university chatbot (UNIBOT). In *2019 3rd international conference on electronics, communication and aerospace technology (ICECA) 2019 Jun 12* (pp. 148-150). IEEE.

Pinto A, Gonçalo Oliveira H, & Oliveira Alves A. Comparing the performance of different NLP toolkits in formal and social media text. In *5th Symposium on Languages, Applications and Technologies (SLATE'16) 2016*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik

Ranoliya BR, Raghuwanshi N, & Singh S. *Chatbot for university related FAQs*. In *2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI) 2017 Sep 13* (pp. 1525-1530). IEEE.

Truong A, Walters A, Goodsitt J, Hines K, Bruss CB, & Farivar R. Towards automated machine learning: Evaluation and comparison of AutoML approaches and tools. In *2019 IEEE 31st international conference on tools with artificial intelligence (ICTAI) 2019 Nov 4* (pp. 1471-1479). IEEE.

Journal

Abdul-Kader SA, & Woods JC. *Survey on chatbot design techniques in speech conversation systems*. **International Journal of Advanced Computer Science and Applications**. 2015;6(7).

Ghadhban D, & Al-Twairish N. *Nabiha: an Arabic dialect chatbot*. **International Journal of Advanced Computer Science and Applications**. 2020;11(3).

- Kaczorowska-Spychalska D. Chatbots in marketing. *Management*. 2019;23(1).
- Kapočiūtė-Dzikiėnė J. A domain-specific generative chatbot trained from little data. *Applied Sciences*. 2020 Jan;10(7):2221.
- Khadija A, Zahra FF, & Naceur A. AI-Powered Health Chatbots: Toward a general architecture. *Procedia Computer Science*. 2021 Jan 1;191:355-60.
- Khanna A, Pandey B, Vashishta K, Kalia K, Pradeepkumar B, & Das T. A study of today's AI through chatbots and rediscovery of machine intelligence. **International Journal of u-and e-Service, Science and Technology**. 2015 Jul 31;8(7):277-84.
- Kraus D, Reibenspiess V, & Eckhardt A. How voice can change customer satisfaction: a comparative analysis between e-commerce and voice commerce.
- Kretzschmar K, Tyroll H, Pavarini G, Manzini A, & Singh I, NeurOx Young People's Advisory Group. Can your phone be your therapist? Young people's ethical perspectives on the use of fully automated conversational agents (chatbots) in mental health support. *Biomedical informatics insights*. 2019 Feb;11:1178222619829083
- Kudo T, & Richardson J. Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. *arXiv preprint arXiv:1808.06226*. 2018 Aug 19
- Kumari P, & Toshniwal D. Long short term memory–convolutional neural network based deep hybrid approach for solar irradiance forecasting. *Applied Energy*. 2021 Aug 1;295:117061.
- Law R, Leung R, Lo A, Leung D, & Fong LH. Distribution channel in hospitality and tourism: Revisiting disintermediation from the perspectives of hotels and travel agencies. **International Journal of Contemporary Hospitality Management**. 2015 Apr 13.
- Le XH, Ho HV, Lee G, & Jung S. Application of long short-term memory (LSTM) neural network for flood forecasting. *Water*. 2019 Jul;11(7):1387.
- Liu CW, Lowe R, Serban IV, Noseworthy M, Charlin L, & Pineau J. How not to evaluate your dialogue system: An empirical study of unsupervised evaluation metrics for dialogue response generation. *arXiv preprint arXiv:1603.08023*. 2016 Mar 25
- McTear M. Conversational AI: dialogue systems, conversational agents, and chatbots. *Synthesis Lectures on Human Language Technologies*. 2020 Oct 30;13(3):1-251.
- McTear MF, Callejas Z, & Griol D. *The conversational interface*. Cham: Springer; 2016.
- Melián-González S, Gutiérrez-Taño D, & Bulchand-Gidumal J. Predicting the intentions to use chatbots for travel and tourism. *Current Issues in Tourism*. 2021 Jan 17;24(2):192-210.

- Merdivan E, Singh D, Hanke S, & Holzinger A. Dialogue systems for intelligent human computer interactions. *Electronic Notes in Theoretical Computer Science*. 2019 May 4;343:57-71.
- Mohialden YM, Younis MT, & Hussien NM. A Novel Approach to Arabic Chabot, Utilizing Google Colab and the Internet of Things: A Case Study at a Computer Center. *Webology*. 2021 Dec;18(2).
- Motger Q, Franch X, & Marco J. Conversational Agents in Software Engineering: Survey, Taxonomy and Challenges. *arXiv preprint arXiv:2106.10901*. 2021 Jun 21.
- Naik NP. Performance Measurement of Natural Dialog System by Analyzing the Conversation. In *Designing User Interfaces With a Data Science Approach 2022* (pp. 180-209). IGI Global
- Ngai EW, Lee MC, Luo M, Chan PS, & Liang T. An intelligent knowledge-based chatbot for customer service. *Electronic Commerce Research and Applications*. 2021 Nov 1;50:101098.
- Nursetyo A, & Subhiyakto ER. Smart chatbot system for e-commerce assistance based on AIML. In *2018 International Seminar on Research of Information Technology and Intelligent Systems (ISRITI) 2018 Nov 21* (pp. 641-645). IEEE.
- Nuruzzaman M, & Hussain OK. A survey on chatbot implementation in customer service industry through deep neural networks. In *2018 IEEE 15th International Conference on e-Business Engineering (ICEBE) 2018 Oct 12* (pp. 54-61). IEEE
- Nuruzzaman M, & Hussain OK. IntelliBot: A Dialogue-based chatbot for the insurance industry. *Knowledge-Based Systems*. 2020 May 21;196:105810.
- Nwankwo W. Interactive advising with bots: improving academic excellence in educational establishments. Wilson Nwankwo. *Interactive Advising with Bots: Improving Academic Excellence in Educational Establishments*. **American Journal of Operations Management and Information Systems**. 2018 Mar 20;3(1):6-21.
- Oguntosin V, & Olomo A. Development of an e-commerce chatbot for a university shopping mall. *Applied Computational Intelligence and Soft Computing*. 2021 Mar 20;2021.
- Oklesinski D, & Jaworski W. Implementation of current state-of-the-art technology for obtaining a working Dialogue Agent.
- Palanica A, Flaschner P, Thommandram A, Li M, & Fossat Y. Physicians' perceptions of chatbots in health care: cross-sectional web-based survey. **Journal of medical Internet research**. 2019 Apr 5;21(4):e12887
- Paliwal S, Bharti V, & Mishra AK. Ai chatbots: Transforming the digital world. In *Recent Trends and Advances in Artificial Intelligence and Internet of Things 2020* (pp. 455-482). Springer, Cham.

- Pardeshi S, Ovhal S, Shinde P, Bansode M, & Birajdar A. A survey on Different Algorithms used in Chatbot. **International Research Journal of Engineering and Technology (IRJET)**. 2020;7:6092-8
- R Khandale, S. Sombansi, S. Mishra, M. F. Shaikh, & P. Mishra, "E-negotiator chatbot for e-commerce websites: implementation," **Journal of Applied Science and Computations**, vol. 6, no. 6, p. 489, 2019
- Rainie H, Anderson JQ, & Albright J. The future of free speech, trolls, anonymity and fake news online. Washington, DC: Pew Research Center; 2017 Mar 29.
- Ram A, Prasad R, Khatri C, Venkatesh A, Gabriel R, Liu Q, Nunn J, Hedayatnia B, Cheng M, Nagar A, & King E. Conversational ai: The science behind the alexa prize. arXiv preprint arXiv:1801.03604. 2018 Jan 11.
- Rana R. Gated recurrent unit (GRU) for emotion classification from noisy speech. arXiv preprint arXiv:1612.07778. 2016 Dec 13.
- Richer R, Zhao N, Eskofier BM, & Paradiso JA. Exploring smart agents for the interaction with multimodal mediated environments. *Multimodal Technologies and Interaction*. 2020 Jun;4(2):27.
- Rutschi C, & Dibbern J. Towards an Understanding of Scaling the Software Robot Implementation. In *Information Systems Outsourcing 2020* (pp. 453-466). Springer, Cham.
- Schachner T, Keller R, & von Wangenheim F. Artificial intelligence-based conversational agents for chronic conditions: systematic literature review. **Journal of medical Internet research**. 2020 Sep 14;22(9):e20701
- Schanke S, Burtch G, & Ray G. Estimating the impact of "humanizing" customer service chatbots. *Information Systems Research*. 2021 Sep;32(3):736-51
- Schuetzler RM, Grimes GM, Giboney JS, & Rosser HK. Deciding Whether and How to Deploy Chatbots. *MIS Quarterly Executive*. 2021 Mar 1;20(1).
- Ścibiorska-Kowalczyk I, & Cichoń J. Kultura, wiedza i informacja w gospodarce cyfrowej. *Pedagogika i zarządzanie*.:169.
- Sennrich R, Haddow B, & Birch A. Neural machine translation of rare words with subword units. arXiv preprint arXiv:1508.07909. 2015 Aug 31.
- Shi N, Zeng Q, & Lee R. The design and implementation of Language Learning Chatbot with XAI using Ontology and Transfer Learning. arXiv preprint arXiv:2009.13984. 2020 Sep 29.
- Shi N, Zeng Q, & Lee RS. The Design and Implementation of Intelligent English Learning Chabot based on Transfer Learning Technology

- Shum HY, He XD, & Li D. From Eliza to XiaoIce: challenges and opportunities with social chatbots. *Frontiers of Information Technology & Electronic Engineering*. 2018 Jan;19(1):10-26
- Singh P, & Kaur A. A systematic review of artificial intelligence in agriculture. *Deep Learning for Sustainable Agriculture*. 2022 Jan 1:57-80.
- Smutny P, & Schreiberova P. Chatbots for learning: A review of educational chatbots for the Facebook Messenger. *Computers & Education*. 2020 Jul 1;151:103862.
- Soufyane A, Abdelhakim BA, & Ahmed MB. An intelligent chatbot using NLP and TF-IDF algorithm for text understanding applied to the medical field. In *Emerging Trends in ICT for Sustainable Development 2021* (pp. 3-10). Springer, Cham.
- Steinbauer F, Kern R, & Kröll M. Chatbots assisting German business management applications. In *International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems 2019* Jul 9 (pp. 717-729). Springer, Cham.
- Subramanian V, Ramachandra N, & Dubash N. TutorBot: contextual learning guide for software engineers. In *2019 IEEE/ACM 1st International Workshop on Bots in Software Engineering (BotSE) 2019* May 28 (pp. 16-17). IEEE
- Tavichaiyuth N, & Rattagan E. Developing Chatbots in Higher Education: A Case Study of Academic Program Chatbot in Thailand.
- Thirumuruganathan S, Tang N, Ouzzani M, & Doan A. Data curation with deep learning [vision]. arXiv preprint arXiv:1803.01384. 2018 Mar 4.
- Trivedi A, Gor V, & Thakkar Z. Chatbot generation and integration: A review. **International Journal of Advance Research**, Ideas and Innovations in Technology. 2019;5(2):1308-11
- VanderPlas J. Python data science handbook: Essential tools for working with data. " O'Reilly Media, Inc."; 2016 Nov 21.
- Vegesna A, Jain P, & Porwal D. Ontology based chatbot (for e-commerce website). **International Journal of Computer Applications**. 2018 Jan;179(14):51-5.
- Wang L, Wang D, Tian F, Peng Z, Fan X, Zhang Z, Yu M, Ma X, & Wang H. Cass: Towards building a social-support chatbot for online health community. *Proceedings of the ACM on Human-Computer Interaction*. 2021 Apr 22;5(CSCW1):1-31.
- Weissensteiner AA. Chatbots as an approach for a faster enquiry handling process in the service industry. *Signature*. 2018;12(04)
- Winkler R, & Söllner M. Unleashing the potential of chatbots in education: A state-of-the-art analysis. In *Academy of Management Annual Meeting (AOM) 2018*.
- Xu B, & Zhuang Z. Survey on psychotherapy chatbots. *Concurrency and Computation: Practice and Experience*. 2022 Mar 25;34(7):e6170.

Yun Y, Ma D, & Yang M. Human-computer interaction-based decision support system with applications in data mining. *Future Generation Computer Systems*. 2021 Jan 1;114:285-9

Zhang Y, Xu L, Mendoza A, Yang G, Chinprutthiwong P, & Gu G. Life after speech recognition: Fuzzing semantic misinterpretation for voice assistant applications. In *Proc. of the Network and Distributed System Security Symposium (NDSS'19)* 2019 Jan.

Zhou L, Gao J, Li D, & Shum HY. The design and implementation of xiaoice, an empathetic social chatbot. *Computational Linguistics*. 2020 Mar 1;46(1):53-93.

Thesis

Akcora C. The effect of chatbots tone of voice on trust and customer satisfaction (Master's thesis, University of Twente).

Dumpala S. *Check-It-Chatbot* (Doctoral dissertation, Dublin Business School).2021

Kockx AJ. Development and Evaluation of a Diagnosis and Triage Healthcare Chatbot (Master's thesis)

Peters F. Master thesis: Design and implementation of a chatbot in the context of customer support. *Design and implementation of a chatbot in the context of customer support*. Masters Dissertation.2018

Sousa FP. *The impact of social virtual presence agents and content-based product recommendation system on on-line customer purchase intention* (Doctoral dissertation).

Tham ZJ. *A conversational question and answer chatbot for university* (Doctoral dissertation, UTAR).

Website

www.javatpoint.com/compatibility-testing

Appendix

Design Code

```
package com.example.chat;

import org.springframework.boot.builder.SpringApplicationBuilder;
import org.springframework.boot.web.servlet.support.SpringBootServletInitializer;

public class ServletInitializer extends SpringBootServletInitializer {

    @Override
    protected SpringApplicationBuilder configure(SpringApplicationBuilder application) {
        return application.sources(Application.class);
    }

}

package com.example.chat;

import org.alicebot.ab.Bot;
import org.apache.commons.io.filefilter.FileFilterUtils;
import org.springframework.beans.factory.annotation.Value;
import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;
import org.springframework.boot.context.event.ApplicationReadyEvent;
import org.springframework.cloud.openfeign.EnableFeignClients;
import org.springframework.context.annotation.Bean;
import org.springframework.context.event.EventListener;
import org.springframework.web.client.RestTemplate;

import java.io.File;
import java.io.FileFilter;
import java.util.Arrays;
import java.util.concurrent.Executors;
import java.util.concurrent.ScheduledExecutorService;
```

```

/**
 * The entry point of the Spring Boot application.
 */
@SpringBootApplication
@EnableFeignClients
public class Application {

    private final String botPath;

    public Application(@Value("${bot.path}") String botPath) {
        this.botPath = botPath;
    }

    public static void main(String[] args) {
        SpringApplication.run(Application.class, args);
    }

    @Bean
    public ScheduledExecutorService executorService() {
        return Executors.newScheduledThreadPool(10);
    }

    @Bean
    public RestTemplate rest() {
        return new RestTemplate();
    }

    @EventListener
    public void createBots(ApplicationReadyEvent event) {
        File[] directories = new File(botPath + "/bots")
            .listFiles((FileFilter) FileFilterUtils.directoryFileFilter());

        if (directories != null) {
            Arrays.stream(directories)
                .map(File::getName)
                .map(name -> new Bot(name,
                    botPath))
                .forEach(bot ->
event.getApplicationContext().getBeanFactory().registerSingleton(bot.name, bot));
        }
    }
}

```

```
}  
}  
  
}
```

```
package com.example.chat.views.main;
```

```
import com.example.chat.views.chat.ChatView;  
import com.example.chat.views.join.JoinView;  
import com.vaadin.flow.component.Text;  
import com.vaadin.flow.component.UI;  
import com.vaadin.flow.component.applayout.AppLayout;  
import com.vaadin.flow.component.applayout.DrawerToggle;  
import com.vaadin.flow.component.dependency.CssImport;  
import com.vaadin.flow.component.html.*;  
import com.vaadin.flow.component.orderedlayout.VerticalLayout;  
import com.vaadin.flow.component.page.Push;  
import com.vaadin.flow.component.tabs.Tab;  
import com.vaadin.flow.component.tabs.Tabs;  
import com.vaadin.flow.component.tabs.TabsVariant;  
import com.vaadin.flow.router.AfterNavigationEvent;  
import com.vaadin.flow.router.AfterNavigationObserver;  
import com.vaadin.flow.router.RouterLink;  
import com.vaadin.flow.server.PWA;  
import com.vaadin.flow.server.VaadinSession;  
import com.vaadin.flow.theme.Theme;  
import com.vaadin.flow.theme.lumo.Lumo;  
import org.alicebot.ab.Bot;  
import org.vaadin.artur.Avataaar;
```

```
import java.util.HashMap;  
import java.util.List;  
import java.util.Map;  
import java.util.stream.Collectors;
```

```
@CssImport("./styles/shared-styles.css")  
@PWA(name = "Tunde AI chat", shortName = "Tunde AI chat")  
@Theme(value = Lumo.class, variant = Lumo.DARK)  
@Push  
public class MainView extends AppLayout implements AfterNavigationObserver {
```

```

private Span botNameContainer = new Span(new Text(""));
private Tabs tabs = new Tabs();
private Map<Tab, String> tabToBotNameMap = new HashMap<>();
private Map<String, Tab> botNameToTabMap = new HashMap<>();
private List<Bot> bots;

public MainView(List<Bot> bots) {
    if (VaadinSession.getCurrent().getAttribute("nickname") == null) {
        UI.getCurrent().navigate(JoinView.class);
        UI.getCurrent().getPage().reload();
        return;
    }

    this.bots = bots;
    setPrimarySection(Section.DRAWER);
    addToNavbar(new DrawerToggle(), botNameContainer);
    createMenuTabs();

    Div footer = new Div();
    footer.addClassName("footer");
    VerticalLayout menu = new VerticalLayout(new H3("Bots:"), tabs, footer);
    menu.addClassName("menu");
    addToDrawer(menu);
}

private void createMenuTabs() {
    tabs.setOrientation(Tabs.Orientation.VERTICAL);
    tabs.addThemeVariants(TabsVariant.LUMO_MINIMAL);
    tabs.setId("tabs");
    tabs.add(getAvailableTabs());
    tabs.addSelectedChangeListener(event ->
setBotName(tabToBotNameMap.get(event.getSelectedTab())));
}

private void setBotName(String botName) {
    botNameContainer.removeAll();
    botNameContainer.add(new Text("Chat with " + botName));
}

private Tab[] getAvailableTabs() {
    return bots.stream()

```

```

        .map(i -> i.name)
        .sorted()
        .map(this::createTab)
        .collect(Collectors.toList()).toArray(new Tab[bots.size()]);
    }

```

```

private Tab createTab(String botName) {
    final Tab tab = new Tab();
    Avataaar avataaar = new Avataaar(botName);
    RouterLink link = new RouterLink(botName, ChatView.class, botName);
    link.addClassName("bot-link");
    tab.add(avataaar, link);
    tabToBotNameMap.put(tab, botName);
    botNameToTabMap.put(botName, tab);
    return tab;
}

```

```

@Override
public void afterNavigation(AfterNavigationEvent event) {
    super.afterNavigation();
    ChatView view = (ChatView) getContent();
    String botName = view.getBotName();
    tabs.setSelectedTab(botNameToTabMap.get(botName));
    setBotName(botName);
}
}

```

```

package com.example.chat.views.join;

```

```

import com.example.chat.views.chat.ChatView;
import com.vaadin.flow.component.Key;
import com.vaadin.flow.component.UI;
import com.vaadin.flow.component.button.Button;
import com.vaadin.flow.component.dependency.CssImport;
import com.vaadin.flow.component.html.H1;
import com.vaadin.flow.component.notification.Notification;
import com.vaadin.flow.component.orderedlayout.VerticalLayout;
import com.vaadin.flow.component.textfield.TextField;
import com.vaadin.flow.router.PageTitle;
import com.vaadin.flow.router.Route;

```

```
import com.vaadin.flow.router.RouteAlias;
import com.vaadin.flow.server.VaadinSession;
```

```
@Route("join")
@CssImport("styles/views/join/join-view.css")
@RouteAlias(value = "")
@PageTitle("Tunde AI Chat")
public class JoinView extends VerticalLayout {
```

```
    public JoinView() {
        addClassName(getClass().getSimpleName());
```

```
        H1 title = new H1("AI Chat");
        title.addClassName(getClass().getSimpleName() + "-title");
```

```
        TextField nickname = new TextField();
        nickname.addClassName(getClass().getSimpleName() + "-nickname");
        nickname.setPlaceholder("Enter your nickname...");
```

```
        Button enter = new Button("Enter", event -> enter(nickname.getValue()));
        enter.addClassName(getClass().getSimpleName() + "-enter");
        enter.addClickShortcut(Key.ENTER);
```

```
        VerticalLayout form = new VerticalLayout(title, nickname, enter);
        form.setSizeUndefined();
        form.addClassName(getClass().getSimpleName() + "-form");
        add(form);
    }
```

```
    private void enter(String nickname) {
        if (nickname.trim().isEmpty()) {
            Notification.show("Enter a nickname");
        } else {
            VaadinSession.getCurrent().setAttribute("nickname", nickname);
            UI.getCurrent().navigate(ChatView.class, "Alice");
        }
    }
}
```

```
}
```

```
package com.example.chat.views.chat;
```

```
import com.vaadin.flow.component.Html;  
import com.vaadin.flow.component.Text;  
import com.vaadin.flow.component.html.Div;  
import com.vaadin.flow.component.html.Span;  
import org.vaadin.artur.Avataaar;
```

```
public class MessageList extends Div {
```

```
    private final int MESSAGE_LIMIT = 50;  
    private Div gap = new Div();
```

```
    public MessageList() {  
        setClassName(getClass().getSimpleName());  
        gap.addClassName(getClass().getSimpleName() + "-gap");  
        add(gap);  
    }
```

```
    public void addMessage(String from, Avataaar avatar, String text, boolean isCurrentUser) {  
        if (getChildren().count() >= MESSAGE_LIMIT + 1) {  
            remove(getChildren().findFirst().get());  
        }  
    }
```

```
        Span fromContainer = new Span(new Text(from));  
        fromContainer.addClassName(getClass().getSimpleName() + "-name");
```

```
        Div textContainer = new Div(new Html("<span>" + text + "</span>"));  
        textContainer.addClassName(getClass().getSimpleName() + "-bubble");
```

```
        Div avatarContainer = new Div(avatar, fromContainer);  
        avatarContainer.addClassName(getClass().getSimpleName() + "-avatar");
```

```
        Div line = new Div(avatarContainer, textContainer);  
        line.addClassName(getClass().getSimpleName() + "-row");  
        add(line);
```

```

if (isCurrentUser) {
    line.addClassName(getClass().getSimpleName() + "-row-currentUser");
    textContainer.addClassName(getClass().getSimpleName() + "-bubble-currentUser");
} else {
    line.addClassName(getClass().getSimpleName() + "-row-otherUser");
    textContainer.addClassName(getClass().getSimpleName() + "-bubble-otherUser");
}

remove(gap);
add(gap);
gap.getElement().callJsFunction("scrollIntoView");
}

public void clear() {
    removeAll();
}

}

package com.example.chat.views.chat;

import com.example.chat.utils.TranslationUtil;
import com.example.chat.views.main.MainView;
import com.vaadin.flow.component.AttachEvent;
import com.vaadin.flow.component.Key;
import com.vaadin.flow.component.UI;
import com.vaadin.flow.component.button.Button;
import com.vaadin.flow.component.dependency.CssImport;
import com.vaadin.flow.component.icon.VaadinIcon;
import com.vaadin.flow.component.orderedlayout.HorizontalLayout;
import com.vaadin.flow.component.orderedlayout.VerticalLayout;
import com.vaadin.flow.component.textfield.TextField;
import com.vaadin.flow.router.BeforeEvent;
import com.vaadin.flow.router.HasUrlParameter;
import com.vaadin.flow.router.PageTitle;
import com.vaadin.flow.router.Route;
import com.vaadin.flow.server.VaadinSession;
import com.vaadin.flow.shared.communication.PushMode;
import org.alicebot.ab.Bot;

```

```

import org.alicebot.ab.Chat;
import org.springframework.context.ApplicationContext;
import org.vaadin.artur.Avataaar;

import java.util.Random;
import java.util.concurrent.ScheduledExecutorService;
import java.util.concurrent.TimeUnit;

@Route(value = "chat", layout = MainView.class)
@PageTitle("Tunde's AI Chat")
@CssImport("styles/views/chat/chat-view.css")
public class ChatView extends VerticalLayout implements HasUrlParameter<String> {

    private final ApplicationContext applicationContext;
    private final ScheduledExecutorService executorService;
    private final UI ui;
    private final MessageList messageList = new MessageList();
    private final TextField message = new TextField();
    private Bot bot;
    private Chat chatSession;

    public ChatView(ApplicationContext applicationContext, ScheduledExecutorService
executorService) {
        this.applicationContext = applicationContext;
        this.executorService = executorService;
        ui = UI.getCurrent();

        message.setPlaceholder("Enter a message...");
        message.setSizeFull();

        Button send = new Button(VaadinIcon.ENTER.create(), event -> sendMessage());
        send.addClickShortcut(Key.ENTER);

        HorizontalLayout inputLayout = new HorizontalLayout(message, send);
        inputLayout.addClassName("inputLayout");

        add(messageList, inputLayout);
        expand(messageList);
        setSizeFull();

```

```

}

@Override
protected void onAttach(AttachEvent attachEvent) {
    ui.getPushConfiguration().setPushMode(PushMode.AUTOMATIC);
}

private void sendMessage() {
    String input = message.getValue();
    TranslationUtil translationUtil = applicationContext.getBean(TranslationUtil.class);
    // get language
    String language = translationUtil.getLanguageName(input);

    // now convert the input to english
    String text = translationUtil.translateToEnglish(language, input);
    // String text = input;
    if (!text.trim().isEmpty()) {
        messageList.addMessage("You", new
Avataaar(VaadinSession.getCurrent().getAttribute("nickname").toString()), text, true);
        message.clear();

        executorService.schedule() -> {
            String answer = chatSession.multisentenceRespond(text);
            ui.access() -> messageList.addMessage(bot.name, new Avataaar(bot.name),
answer.isEmpty() ? "... " : translationUtil.translateFromEnglish(answer, language), false));
            }, new Random().ints(1000, 3000).findFirst().getAsInt(), TimeUnit.MILLISECONDS);
        }
    }
}

@Override
public void setParameter(BeforeEvent event, String botName) {
    bot = (Bot) applicationContext.getBean(botName);
    chatSession = new Chat(bot);
    messageList.clear();
}

public String getBotName() {
    return bot.name;
}

```

```
}
```

```
package com.example.chat.utils;
```

```
import com.detectlanguage.DetectLanguage;  
import com.detectlanguage.Result;  
import com.detectlanguage.errors.APIError;  
import org.springframework.beans.factory.annotation.Value;  
import org.springframework.http.HttpEntity;  
import org.springframework.http.HttpHeaders;  
import org.springframework.http.HttpMethod;  
import org.springframework.http.MediaType;  
import org.springframework.stereotype.Service;  
import org.springframework.web.client.RestTemplate;
```

```
import java.util.ArrayList;  
import java.util.Collections;  
import java.util.List;
```

```
@Service
```

```
public class TranslationUtil {
```

```
    @Value("${app.detection.api.key:905e766c080c733644f3cdf371434568}")  
    private String detectionAPIKey;
```

```
    @Value("${google.translate.url}")  
    private String url;
```

```
    private final RestTemplate restTemplate;  
    private final GoogleApiClient googleApiClient;
```

```
    public TranslationUtil(RestTemplate restTemplate, GoogleApiClient googleApiClient) {  
        this.restTemplate = restTemplate;
```

```
        this.googleApiClient = googleApiClient;  
    }  
}
```

```

public String translateFromEnglish(String word, String preferredLanguage) {
    if ("en".equalsIgnoreCase(preferredLanguage)) {
        return word;
    }
    TranslationData translationData = new TranslationData();
    translationData.setOriginal(word);

    TranslationRequest request = new TranslationRequest();
    request.setTo(preferredLanguage);
    request.setFrom("EN");
    request.setData(Collections.singletonList(translationData));

    TranslationResponse translationResponse = getResponse(request);
    return translationResponse.getData().get(0).getTranslated();
}

```

```

public String translateToEnglish(String initialLanguage, String word) {
    if ("en".equalsIgnoreCase(initialLanguage)) {
        return word;
    }
    TranslationData translationData = new TranslationData();
    translationData.setOriginal(word);

    TranslationRequest request = new TranslationRequest();
    request.setTo("EN");
    request.setFrom(initialLanguage);
    request.setData(Collections.singletonList(translationData));

    TranslationResponse translationResponse = getResponse(request);
    return translationResponse.getData().get(0).getTranslated();
}

```

```

public String getLanguageName(String word) {
    List<Result> results = new ArrayList<>();
    try {
        DetectLanguage.apiKey = detectionAPIKey;
        results = DetectLanguage.detect(word);
    } catch (APIError e) {
        e.printStackTrace();
    }
}

```

```

    if (results.isEmpty()) {
        return word;
    }

    Result result = results.get(0);

    System.out.println("Language: " + result.language);
    System.out.println("Is reliable: " + result.isReliable);
    System.out.println("Confidence: " + result.confidence);

    return result.language;
}

private HttpHeaders getHeader() {
    HttpHeaders httpHeaders = new HttpHeaders();
    httpHeaders.setAccept(Collections.singletonList(MediaType.APPLICATION_JSON));
    httpHeaders.set("User-Agent", "Mozilla/5.0");
    httpHeaders.set("Host", "developer.mozilla.org");
    httpHeaders.set("content-type", MediaType.APPLICATION_JSON.toString());
    return httpHeaders;
}

private TranslationResponse getResponse(TranslationRequest request) {
    try {
        HttpEntity<TranslationRequest> entity = new HttpEntity<>(request, getHeader());
        return restTemplate.exchange(url, HttpMethod.POST, entity,
TranslationResponse.class).getBody();
    } catch (Exception e) {
        e.printStackTrace();
        TranslationResponse translationResponse = new TranslationResponse();
        translationResponse.setData(request.getData());
        return translationResponse;
    }
}
}
}

```

```
package com.example.chat.utils;
```

```
import com.fasterxml.jackson.annotation.JsonIgnoreProperties;  
import com.fasterxml.jackson.annotation.JsonInclude;
```

```
import java.util.ArrayList;  
import java.util.List;
```

```
@JsonIgnoreProperties(ignoreUnknown = true)  
@JsonInclude(JsonInclude.Include.NON_EMPTY)  
public class TranslationResponse {  
    private List<TranslationData> data = new ArrayList<>();  
    private String message;
```

```
    public List<TranslationData> getData() {  
        return data;  
    }
```

```
    public void setData(List<TranslationData> data) {  
        this.data = data;  
    }
```

```
    public String getMessage() {  
        return message;  
    }
```

```
    public void setMessage(String message) {  
        this.message = message;  
    }  
}
```

```
package com.example.chat.utils;
```

```
import com.fasterxml.jackson.annotation.JsonIgnoreProperties;  
import com.fasterxml.jackson.annotation.JsonInclude;
```

```
import java.util.*;
```

```

@JsonIgnoreProperties(ignoreUnknown = true)
@JsonInclude(JsonInclude.Include.NON_EMPTY)
public class TranslationRequest {

    private List<TranslationData> data = new ArrayList<>();
    private String from;
    private String to;

    public List<TranslationData> getData() {
        return data;
    }

    public void setData(List<TranslationData> data) {
        this.data = data;
    }

    public String getFrom() {
        return from;
    }

    public void setFrom(String from) {
        this.from = from;
    }

    public String getTo() {
        return to;
    }

    public void setTo(String to) {
        this.to = to;
    }
}

package com.example.chat.utils;

import com.fasterxml.jackson.annotation.JsonIgnoreProperties;

```

```
import com.fasterxml.jackson.annotation.JsonInclude;
```

```
@JsonIgnoreProperties(ignoreUnknown = true)  
@JsonInclude(JsonInclude.Include.NON_EMPTY)
```

```
public class TranslationData {  
    private String original;  
    private String translated;
```

```
  
    public String getTranslated() {  
        return translated;  
    }  
}
```

```
  
    public void setTranslated(String translated) {  
        this.translated = translated;  
    }  
}
```

```
  
    public String getOriginal() {  
        return original;  
    }  
}
```

```
  
    public void setOriginal(String original) {  
        this.original = original;  
    }  
}
```

```
package com.example.chat.utils;
```

```
import feign.RequestInterceptor;  
import feign.RequestTemplate;  
import lombok.extern.slf4j.Slf4j;  
import org.springframework.http.MediaType;
```

```
@Slf4j  
public class GoogleFeignClientInterceptor implements RequestInterceptor {
```

```
  
    @Override  
    public void apply(RequestTemplate template) {
```

```

        template.header("User-Agent", "Mozilla/5.0");
        template.header("Host", "developer.mozilla.org");
        template.header("content-type", MediaType.APPLICATION_JSON.toString());
    }
}

package com.example.chat.utils;

import org.springframework.context.annotation.Bean;

public class GoogleConfig {
    @Bean
    public GoogleFeignClientInterceptor googleFeignClientInterceptor() {
        return new GoogleFeignClientInterceptor();
    }
}

package com.example.chat.utils;

import org.springframework.cloud.openfeign.FeignClient;
import org.springframework.web.bind.annotation.PostMapping;
import org.springframework.web.bind.annotation.RequestBody;

@FeignClient(name = "google-service", url = "${google.translate.url}", configuration =
GoogleConfig.class)
public interface GoogleApiClient {

    @PostMapping("")
    TranslationResponse translate(

        @RequestBody TranslationRequest body

```

```

);

}
package com.example.chat.utils;

import feign.Logger;
import org.springframework.cloud.openfeign.EnableFeignClients;
import org.springframework.cloud.openfeign.FeignClientsConfiguration;
import org.springframework.context.annotation.Bean;
import org.springframework.context.annotation.Configuration;
import org.springframework.context.annotation.Import;

```

```

@Configuration
@EnableFeignClients(basePackages = "com.example.chat")
@Import(FeignClientsConfiguration.class)
public class FeignConfiguration {

    /**
     * Set the Feign specific log level to log client REST requests.
     */
    @Bean
    Logger.Level feignLoggerLevel() {
        return Logger.Level.FULL;
    }

}

```

```

<?xml version="1.0" encoding="ISO-8859-1"?>
<aiml version="1.0">
  <category>
    <pattern>CAN I OPEN AN ACCOUNT WHILE OUTSIDE THE COUNTRY</pattern>
    <template>You can open an account online here while out of the country if you already have a
Bank Verification Number (BVN). If you do not have a BVN, you will still receive an account
number, but the account will remain restricted until you complete the required
documentation.</template>
  </category>
  <category>
    <pattern>IS IT POSSIBLE TO OPEN AN ACCOUNT IN THE COUNTRY AND OPERATE
THE ACCOUNT WHILE OUT OF THE COUNTRY</pattern>

```

<template>Yes. The account can be operated through bank card products or internet banking solution.</template>

</category>

<category>

<pattern>HOW CAN I OPEN A SALARY ACCOUNT</pattern>

<template>Your company / organisation will write to bank to introduce the prospective account holder to the bank for account opening.</template>

</category>

<category>

<pattern>HOW LONG DOES IT TAKE BEFORE MY ACCOUNT BECOMES INACTIVE</pattern>

<template>Current accounts become dormant after 6 months of inactivity.</template>

</category>

<category>

<pattern>Can A Savings Account Become Dormant</pattern>

<template>No. savings accounts do not go dormant.</template>

</category>

<category>

<pattern>How Long Does It Take To Open An Account In Banks</pattern>

<template>Accounts are opened within 24 hours upon submission of complete documentation.</template>

</category>

<category>

<pattern>What Are The Requirements For A Change Of Address in Banks</pattern>

<template>A duly signed instruction from the account holder informing the bank of the change of address, as well as a current utility bill which has the new address of the customer, are needed for a change of address.</template>

</category>

<category>

<pattern>What is TIN</pattern>

<template>Tax Identification Number. A unique code for recognising a taxpayer on FIRS database.</template>

</category>

<category>

<pattern>Can payment be made in any Bank's branch</pattern>

<template>Yes, you can pay your tax at any Bank's branch.</template>

</category>

<category>

<pattern>What documents required for payment</pattern>

<template>Customer TIN, Assessment Form, WHT schedule.</template>

</category>

<category>

<pattern>How can a new taxpayer get the TIN</pattern>

<template>Taxpayers can get their Tax Identification Number (TIN) from any Federal Inland Revenue Service (FIRS) tax office.</template>

</category>

```

<category>
  <pattern>Must a taxpayer be assessed before payment</pattern>
  <template>Pre-payment assessment is not necessary. However, new taxpayers (individual and
corporate) are assessed before payment of certain taxes, for instance POL Tax, Penalty,
etc.</template>
</category>
<category>
  <pattern>CAN PAYMENT BE MADE AT ANY BANK</pattern>
  <template>Payment must be made at the bank where the Form M was established.</template>
</category>
<category>
  <pattern>WHAT ARE THE DOCUMENTATION REQUIRED FOR PAYMENT</pattern>
  <template>Assessment Notice from Customs.</template>
</category>
<category>
  <pattern>WHAT IS SGD ASSESSMENT NOTICE</pattern>
  <template>Assessment Notice contain details of duty to be paid.</template>
</category>
<category>
  <pattern>WHO ISSUED THE SGD ASSESSMENT NOTICE</pattern>
  <template>Nigeria Customs Service (NCS).</template>
</category>
<category>
  <pattern>WILL I HAVE TO START THE PROCESS FOR MY PASSPORT OR VISA IN
THE BANK</pattern>
  <template>No you will not. Registration is done online via the NIS portal. Only payment is
made in the bank.</template>
</category>
<category>
  <pattern>WILL I GET MY VISA OR PASSPORT ONCE PAYMENT IS MADE IN THE
BANK</pattern>
  <template>Yes, but at the NIS office, after completion of all processes involved. Upon
payment, customer takes his or her proof of payment to the NIS office where the passport or visa
is finally processed.</template>
</category>
</aiml>
}

```

```

private Tab[] getAvailableTabs() {
  return bots.stream()
    .map(i -> i.name)
    .sorted()
    .map(this::createTab)
    .collect(Collectors.toList()).toArray(new Tab[bots.size()]);
}

```

```

}

private Tab createTab(String botName) {
    final Tab tab = new Tab();
    Avataaar avataaar = new Avataaar(botName);
    RouterLink link = new RouterLink(botName, ChatView.class, botName);
    link.addClassName("bot-link");
    tab.add(avataaar, link);
    tabToBotNameMap.put(tab, botName);
    botNameToTabMap.put(botName, tab);
    return tab;
}

@Override
public void afterNavigation(AfterNavigationEvent event) {
    super.afterNavigation();
    ChatView view = (ChatView) getContent();
    String botName = view.getBotName();
    tabs.setSelectedTab(botNameToTabMap.get(botName));
    setBotName(botName);
}

}

package com.example.chat.views.join;

import com.example.chat.views.chat.ChatView;
import com.vaadin.flow.component.Key;
import com.vaadin.flow.component.UI;
import com.vaadin.flow.component.button.Button;
import com.vaadin.flow.component.dependency.CssImport;
import com.vaadin.flow.component.html.H1;
import com.vaadin.flow.component.notification.Notification;
import com.vaadin.flow.component.orderedlayout.VerticalLayout;
import com.vaadin.flow.component.textfield.TextField;
import com.vaadin.flow.router.PageTitle;
import com.vaadin.flow.router.Route;
import com.vaadin.flow.router.RouteAlias;
import com.vaadin.flow.server.VaadinSession;

```

```

@Route("join")
@CssImport("styles/views/join/join-view.css")
@RouteAlias(value = "")
@PageTitle("Tunde AI Chat")
public class JoinView extends VerticalLayout {

    public JoinView() {
        addClassName(getClass().getSimpleName());

        H1 title = new H1("AI Chat");
        title.addClassName(getClass().getSimpleName() + "-title");

        TextField nickname = new TextField();
        nickname.addClassName(getClass().getSimpleName() + "-nickname");
        nickname.setPlaceholder("Enter your nickname...");

        Button enter = new Button("Enter", event -> enter(nickname.getValue()));
        enter.addClassName(getClass().getSimpleName() + "-enter");
        enter.addClickShortcut(Key.ENTER);

        VerticalLayout form = new VerticalLayout(title, nickname, enter);
        form.setSizeUndefined();
        form.addClassName(getClass().getSimpleName() + "-form");
        add(form);
    }

    private void enter(String nickname) {
        if (nickname.trim().isEmpty()) {
            Notification.show("Enter a nickname");
        } else {
            VaadinSession.getCurrent().setAttribute("nickname", nickname);
            UI.getCurrent().navigate(ChatView.class, "Alice");
        }
    }
}

package com.example.chat.views.chat;

```

```
import com.vaadin.flow.component.Html;
import com.vaadin.flow.component.Text;
import com.vaadin.flow.component.html.Div;
import com.vaadin.flow.component.html.Span;
import org.vaadin.artur.Avataaar;
```

```
public class MessageList extends Div {
```

```
    private final int MESSAGE_LIMIT = 50;
    private Div gap = new Div();
```

```
    public MessageList() {
        setClassName(getClass().getSimpleName());
        gap.addClassName(getClass().getSimpleName() + "-gap");
        add(gap);
    }
```

```
    public void addMessage(String from, Avataaar avatar, String text, boolean isCurrentUser) {
        if (getChildren().count() >= MESSAGE_LIMIT + 1) {
            remove(getChildren().findFirst().get());
        }
    }
```

```
        Span fromContainer = new Span(new Text(from));
        fromContainer.addClassName(getClass().getSimpleName() + "-name");
```

```
        Div textContainer = new Div(new Html("<span>" + text + "</span>"));
        textContainer.addClassName(getClass().getSimpleName() + "-bubble");
```

```
        Div avatarContainer = new Div(avatar, fromContainer);
        avatarContainer.addClassName(getClass().getSimpleName() + "-avatar");
```

```
        Div line = new Div(avatarContainer, textContainer);
        line.addClassName(getClass().getSimpleName() + "-row");
        add(line);
```

```
        if (isCurrentUser) {
```

```

        line.addClassName(getClass().getSimpleName() + "-row-currentUser");
        textContainer.addClassName(getClass().getSimpleName() + "-bubble-currentUser");
    } else {
        line.addClassName(getClass().getSimpleName() + "-row-otherUser");
        textContainer.addClassName(getClass().getSimpleName() + "-bubble-otherUser");
    }
}

remove(gap);
add(gap);
gap.getElement().callJsFunction("scrollIntoView");
}

public void clear() {
    removeAll();
}

}

package com.example.chat.views.chat;

import com.example.chat.utils.TranslationUtil;
import com.example.chat.views.main.MainView;
import com.vaadin.flow.component.AttachEvent;
import com.vaadin.flow.component.Key;
import com.vaadin.flow.component.UI;
import com.vaadin.flow.component.button.Button;
import com.vaadin.flow.component.dependency.CssImport;
import com.vaadin.flow.component.icon.VaadinIcon;
import com.vaadin.flow.component.orderedlayout.HorizontalLayout;
import com.vaadin.flow.component.orderedlayout.VerticalLayout;
import com.vaadin.flow.component.textfield.TextField;
import com.vaadin.flow.router.BeforeEvent;
import com.vaadin.flow.router.HasUrlParameter;
import com.vaadin.flow.router.PageTitle;
import com.vaadin.flow.router.Route;
import com.vaadin.flow.server.VaadinSession;
import com.vaadin.flow.shared.communication.PushMode;
import org.alicebot.ab.Bot;
import org.alicebot.ab.Chat;
import org.springframework.context.ApplicationContext;
import org.vaadin.artur.Avataaar;

```

```
import java.util.Random;
import java.util.concurrent.ScheduledExecutorService;
import java.util.concurrent.TimeUnit;
```

```
@Route(value = "chat", layout = MainView.class)
@PageTitle("Tunde's AI Chat")
@CssImport("styles/views/chat/chat-view.css")
public class ChatView extends VerticalLayout implements HasUrlParameter<String> {
```

```
    private final ApplicationContext applicationContext;
    private final ScheduledExecutorService executorService;
    private final UI ui;
    private final MessageList messageList = new MessageList();
    private final TextField message = new TextField();
    private Bot bot;
    private Chat chatSession;
```

```
    public ChatView(ApplicationContext applicationContext, ScheduledExecutorService
executorService) {
```

```
        this.applicationContext = applicationContext;
        this.executorService = executorService;
        ui = UI.getCurrent();
```

```
        message.setPlaceholder("Enter a message...");
        message.setSizeFull();
```

```
        Button send = new Button(VaadinIcon.ENTER.create(), event -> sendMessage());
        send.addClickShortcut(Key.ENTER);
```

```
        HorizontalLayout inputLayout = new HorizontalLayout(message, send);
        inputLayout.addClassName("inputLayout");
```

```
        add(messageList, inputLayout);
        expand(messageList);
        setSizeFull();
```

```
    }
```

```

@Override
protected void onAttach(AttachEvent attachEvent) {
    ui.getPushConfiguration().setPushMode(PushMode.AUTOMATIC);
}

private void sendMessage() {
    String input = message.getValue();
    TranslationUtil translationUtil = applicationContext.getBean(TranslationUtil.class);
    // get language
    String language = translationUtil.getLanguageName(input);

    // now convert the input to english
    String text = translationUtil.translateToEnglish(language, input);
    // String text = input;
    if (!text.trim().isEmpty()) {
        messageList.addMessage("You", new
Avataaar(VaadinSession.getCurrent().getAttribute("nickname").toString()), text, true);
        message.clear();

        executorService.schedule(() -> {
            String answer = chatSession.multisentenceRespond(text);
            ui.access() -> messageList.addMessage(bot.name, new Avataaar(bot.name),
answer.isEmpty() ? "...": translationUtil.translateFromEnglish(answer, language), false));
            }, new Random().ints(1000, 3000).findFirst().getAsInt(), TimeUnit.MILLISECONDS);
        }
    }
}

@Override
public void setParameter(BeforeEvent event, String botName) {
    bot = (Bot) applicationContext.getBean(botName);
    chatSession = new Chat(bot);
    messageList.clear();
}

public String getBotName() {
    return bot.name;
}
}
}

```

```
package com.example.chat.utils;
```

```
import com.detectlanguage.DetectLanguage;  
import com.detectlanguage.Result;  
import com.detectlanguage.errors.APIError;  
import org.springframework.beans.factory.annotation.Value;  
import org.springframework.http.HttpEntity;  
import org.springframework.http.HttpHeaders;  
import org.springframework.http.HttpMethod;  
import org.springframework.http.MediaType;  
import org.springframework.stereotype.Service;  
import org.springframework.web.client.RestTemplate;
```

```
import java.util.ArrayList;  
import java.util.Collections;  
import java.util.List;
```

```
@Service  
public class TranslationUtil {
```

```
    @Value("${app.detection.api.key:905e766c080c733644f3cdf371434568}")  
    private String detectionAPIKey;
```

```
    @Value("${google.translate.url}")  
    private String url;
```

```
    private final RestTemplate restTemplate;  
    private final GoogleApiClient googleApiClient;
```

```
    public TranslationUtil(RestTemplate restTemplate, GoogleApiClient googleApiClient) {  
        this.restTemplate = restTemplate;
```

```
        this.googleApiClient = googleApiClient;  
    }
```

```
    public String translateFromEnglish(String word, String preferredLanguage) {  
        if ("en".equalsIgnoreCase(preferredLanguage)) {  
            return word;
```

```

    }
    TranslationData translationData = new TranslationData();
    translationData.setOriginal(word);

    TranslationRequest request = new TranslationRequest();
    request.setTo(preferredLanguage);
    request.setFrom("EN");
    request.setData(Collections.singletonList(translationData));

    TranslationResponse translationResponse = getResponse(request);
    return translationResponse.getData().get(0).getTranslated();
}

public String translateToEnglish(String initialLanguage, String word) {
    if ("en".equalsIgnoreCase(initialLanguage)) {
        return word;
    }
    TranslationData translationData = new TranslationData();
    translationData.setOriginal(word);

    TranslationRequest request = new TranslationRequest();
    request.setTo("EN");
    request.setFrom(initialLanguage);
    request.setData(Collections.singletonList(translationData));

    TranslationResponse translationResponse = getResponse(request);
    return translationResponse.getData().get(0).getTranslated();
}

public String getLanguageName(String word) {
    List<Result> results = new ArrayList<>();
    try {
        DetectLanguage.apiKey = detectionAPIKey;
        results = DetectLanguage.detect(word);
    } catch (APIError e) {
        e.printStackTrace();
    }

    if (results.isEmpty()) {

```

```

        return word;
    }

    Result result = results.get(0);

    System.out.println("Language: " + result.language);
    System.out.println("Is reliable: " + result.isReliable);
    System.out.println("Confidence: " + result.confidence);

    return result.language;

}

private HttpHeaders getHeader() {
    HttpHeaders httpHeaders = new HttpHeaders();
    httpHeaders.setAccept(Collections.singletonList(MediaType.APPLICATION_JSON));
    httpHeaders.set("User-Agent", "Mozilla/5.0");
    httpHeaders.set("Host", "developer.mozilla.org");
    httpHeaders.set("content-type", MediaType.APPLICATION_JSON.toString());
    return httpHeaders;
}

private TranslationResponse getResponse(TranslationRequest request) {
    try {
        HttpEntity<TranslationRequest> entity = new HttpEntity<>(request, getHeader());
        return restTemplate.exchange(url, HttpMethod.POST, entity,
TranslationResponse.class).getBody();
    } catch (Exception e) {
        e.printStackTrace();
        TranslationResponse translationResponse = new TranslationResponse();
        translationResponse.setData(request.getData());
        return translationResponse;
    }
}

}

package com.example.chat.utils;

```

```
import com.fasterxml.jackson.annotation.JsonIgnoreProperties;
import com.fasterxml.jackson.annotation.JsonInclude;
```

```
import java.util.ArrayList;
import java.util.List;
```

```
@JsonIgnoreProperties(ignoreUnknown = true)
@JsonInclude(JsonInclude.Include.NON_EMPTY)
public class TranslationResponse {
    private List<TranslationData> data = new ArrayList<>();
    private String message;
```

```
    public List<TranslationData> getData() {
        return data;
    }
```

```
    public void setData(List<TranslationData> data) {
        this.data = data;
    }
```

```
    public String getMessage() {
        return message;
    }
```

```
    public void setMessage(String message) {
        this.message = message;
    }
}
```

```
package com.example.chat.utils;
```

```
import com.fasterxml.jackson.annotation.JsonIgnoreProperties;
import com.fasterxml.jackson.annotation.JsonInclude;
```

```
import java.util.*;
```

```

@JsonIgnoreProperties(ignoreUnknown = true)
@JsonInclude(JsonInclude.Include.NON_EMPTY)
public class TranslationRequest {

    private List<TranslationData> data = new ArrayList<>();
    private String from;
    private String to;

    public List<TranslationData> getData() {
        return data;
    }

    public void setData(List<TranslationData> data) {
        this.data = data;
    }

    public String getFrom() {
        return from;
    }

    public void setFrom(String from) {
        this.from = from;
    }

    public String getTo() {
        return to;
    }

    public void setTo(String to) {
        this.to = to;
    }
}

package com.example.chat.utils;

import com.fasterxml.jackson.annotation.JsonIgnoreProperties;
import com.fasterxml.jackson.annotation.JsonInclude;

```

```

@JsonIgnoreProperties(ignoreUnknown = true)
@JsonInclude(JsonInclude.Include.NON_EMPTY)
public class TranslationData {
    private String original;
    private String translated;

    public String getTranslated() {
        return translated;
    }

    public void setTranslated(String translated) {
        this.translated = translated;
    }

    public String getOriginal() {
        return original;
    }

    public void setOriginal(String original) {
        this.original = original;
    }
}

package com.example.chat.utils;

import feign.RequestInterceptor;
import feign.RequestTemplate;
import lombok.extern.slf4j.Slf4j;
import org.springframework.http.MediaType;

@Slf4j
public class GoogleFeignClientInterceptor implements RequestInterceptor {

    @Override
    public void apply(RequestTemplate template) {

        template.header("User-Agent", "Mozilla/5.0");
        template.header("Host", "developer.mozilla.org");
    }
}

```

```

        template.header("content-type", MediaType.APPLICATION_JSON.toString());
    }

}

package com.example.chat.utils;

import org.springframework.context.annotation.Bean;

public class GoogleConfig {
    @Bean
    public GoogleFeignClientInterceptor googleFeignClientInterceptor() {
        return new GoogleFeignClientInterceptor();
    }
}

package com.example.chat.utils;

import org.springframework.cloud.openfeign.FeignClient;
import org.springframework.web.bind.annotation.PostMapping;
import org.springframework.web.bind.annotation.RequestBody;

@FeignClient(name = "google-service", url = "${google.translate.url}", configuration =
GoogleConfig.class)
public interface GoogleApiClient {

    @PostMapping("")
    TranslationResponse translate(

        @RequestBody TranslationRequest body
    );
}

```

```
package com.example.chat.utils;
```

```
import feign.Logger;
import org.springframework.cloud.openfeign.EnableFeignClients;
import org.springframework.cloud.openfeign.FeignClientsConfiguration;
import org.springframework.context.annotation.Bean;
import org.springframework.context.annotation.Configuration;
import org.springframework.context.annotation.Import;
```

```
@Configuration
@EnableFeignClients(basePackages = "com.example.chat")
@Import(FeignClientsConfiguration.class)
public class FeignConfiguration {
```

```
    /**
     * Set the Feign specific log level to log client REST requests.
     */
```

```
    @Bean
    Logger.Level feignLoggerLevel() {
        return Logger.Level.FULL;
    }
```

```
}
```

```
<?xml version="1.0" encoding="ISO-8859-1"?>
```

```
<aiml version="1.0">
```

```
    <category>
```

```
        <pattern>CAN I OPEN AN ACCOUNT WHILE OUTSIDE THE COUNTRY</pattern>
```

```
        <template>You can open an account online here while out of the country if you already have a Bank Verification Number (BVN). If you do not have a BVN, you will still receive an account number, but the account will remain restricted until you complete the required documentation.</template>
```

```
    </category>
```

```
    <category>
```

```
        <pattern>IS IT POSSIBLE TO OPEN AN ACCOUNT IN THE COUNTRY AND OPERATE THE ACCOUNT WHILE OUT OF THE COUNTRY</pattern>
```

```
        <template>Yes. The account can be operated through bank card products or internet banking solution.</template>
```

```
    </category>
```

```
    <category>
```

```
        <pattern>HOW CAN I OPEN A SALARY ACCOUNT</pattern>
```

<template>Your company / organisation will write to bank to introduce the prospective account holder to the bank for account opening.</template>

</category>

<category>

<pattern>HOW LONG DOES IT TAKE BEFORE MY ACCOUNT BECOMES INACTIVE</pattern>

<template>Current accounts become dormant after 6 months of inactivity.</template>

</category>

<category>

<pattern>Can A Savings Account Become Dormant</pattern>

<template>No. savings accounts do not go dormant.</template>

</category>

<category>

<pattern>How Long Does It Take To Open An Account In Banks</pattern>

<template>Accounts are opened within 24 hours upon submission of complete documentation.</template>

</category>

<category>

<pattern>What Are The Requirements For A Change Of Address in Banks</pattern>

<template>A duly signed instruction from the account holder informing the bank of the change of address, as well as a current utility bill which has the new address of the customer, are needed for a change of address.</template>

</category>

<category>

<pattern>What is TIN</pattern>

<template>Tax Identification Number. A unique code for recognising a taxpayer on FIRS database.</template>

</category>

<category>

<pattern>Can payment be made in any Bank's branch</pattern>

<template>Yes, you can pay your tax at any Bank's branch.</template>

</category>

<category>

<pattern>What documents required for payment</pattern>

<template>Customer TIN, Assessment Form, WHT schedule.</template>

</category>

<category>

<pattern>How can a new taxpayer get the TIN</pattern>

<template>Taxpayers can get their Tax Identification Number (TIN) from any Federal Inland Revenue Service (FIRS) tax office.</template>

</category>

<category>

<pattern>Must a taxpayer be assessed before payment</pattern>

<template>Pre-payment assessment is not necessary. However, new taxpayers (individual and corporate) are assessed before payment of certain taxes, for instance POL Tax, Penalty, etc.</template>

</category>
<category>
<pattern>CAN PAYMENT BE MADE AT ANY BANK</pattern>
<template>Payment must be made at the bank where the Form M was established.</template>
</category>
<category>
<pattern>WHAT ARE THE DOCUMENTATION REQUIRED FOR PAYMENT</pattern>
<template>Assessment Notice from Customs.</template>
</category>
<category>
<pattern>WHAT IS SGD ASSESSMENT NOTICE</pattern>
<template>Assessment Notice contain details of duty to be paid.</template>
</category>
<category>
<pattern>WHO ISSUED THE SGD ASSESSMENT NOTICE</pattern>
<template>Nigeria Customs Service (NCS).</template>
</category>
<category>
<pattern>WILL I HAVE TO START THE PROCESS FOR MY PASSPORT OR VISA IN THE BANK</pattern>
<template>No you will not. Registration is done online via the NIS portal. Only payment is made in the bank.</template>
</category>
<category>
<pattern>WILL I GET MY VISA OR PASSPORT ONCE PAYMENT IS MADE IN THE BANK</pattern>
<template>Yes, but at the NIS office, after completion of all processes involved. Upon payment, customer takes his or her proof of payment to the NIS office where the passport or visa is finally processed.</template>
</category>
</aiml>

Biodata

Personal Data:

Surname: AKINYEMI

Other Names: ROQIB AKINTUNDE

Date of Birth: 8th August 1986

Gender: Male

Local Government Area: Atakumosa West

State of Origin: Osun State

Nationality: Nigerian

Marital Status: Married

Religion: Islam

Next of Kin: Fausat Akinyemi

Contact:

Residential Address: 11, Omotola Akinbiyi close, off Sajim Avenue, Daleko Ogunrun Area, Mowe Ogun State.

Postal Address: 45, Idumagbo Avenue, Lagos Island, Lagos State.

Email Address: roqibakinyemi2010@gmail.com

Mobile No: 08070928156

Educational Background:

Institutions attended with Dates and Qualification:

Lead City University, Toll Gate, Ibadan, Oyo State: 2020 till date (MSc. Computer Science)

Crescent University, Abeokuta: 2013-2015 (BSc. Computer Science)

Osun State Polytechnic, Iree: 2008-2010 (HND. Computer Science)

Osun State Polytechnic, Iree: 2004-2007 (ND. Computer Science)

Ansar-ud-Deen High School, Liberty Road, Oke-Ado Ibadan: 1996-2002 (SSCE, O'Level)

St. James Cathedral Primary School & Agbaje Memorial Primary School: 1991-1996 (First School Leaving Certificate)

Work Experience with Dates

Access Bank Plc. Analyst, IT Service Desk - 2012

Polaris Bank Limited, Analyst, Core Banking Application - 2018

First Bank of Nigeria Limited, Information Technology Incident and Problem Manager - 2020

Interswitch Fintech Company - 2022

Membership of Academic Professional Bodies

Graduate Member, Computer Professionals (Registration Council of Nigeria)

Academic Papers (Unpublished These):

Project submission Development of a Website of the Faculty of ICT, Osun State Polytechnic, Iree HND. Thesis, 2010.

Project submission Design and Implementation of Online Examination Application B.Sc. Thesis, 2015.

Published Journal Articles

Title: **Automation of Customer Support System (Chatbot) to Solve Web Based Financial and Payment Application Service**

Year of publication: August 2023

Volume: Vol. 12 No. 2 (2023): July-December 2023

Journal: Asian Journal of Computer Science and Technology

Journal ID: <https://doi.org/10.51983/ajcst-2023.12.2.3697>

Title: **Multi-criteria Decision Support System for Prioritizing Challenges of Internet of Things (IoT)**

Year of publication: August 2023

Volume: 11 Issue VII Jul 2023

Journal: International Journal for Research in Applied Science & Engineering Technology

Journal ID: DOI: 10.22214/ijraset.2023.54873 2.

Title: **Software Project Management: Evaluation of Agile Software Processes a Short Comparison with Waterfall Model**

Year of Publication: November 2021

Volume: 9 Number 10: 115.

Journal: American Journal of Computer Science and Information Technology.

Skills

- i. Structured Query Language
- ii. Power BI
- iii. IT Service Delivery
- iv. Strong Analytical Skills

Referees

1. Mr. Mudathir Adefarati

Accountant/Head, Admin. & Finance

Children's International School, Lagos

08038332043

2. Mr. Ramon Musa

Senior Manager

KPMG Nigeria

07033981596

3. Mr. Mashud momoh

Team Lead, Flexcube Implementation and Support

Tenece Professional Services

0803 448 5062

.....

Signature

.....

Date

Do Not Copy, Lead City University, Nigeria

The University Compliance Certification

This is to certify that this thesis by AKINYEMI ROQIB AKINTUNDE with Matriculation Number LCU/PG/001893 in the Department of Computer Science, Faculty of Natural and Applied Sciences, Lead City University, Ibadan is in full compliance with the approval of the University's format and style.

.....

Signature

.....

Date

Do Not Copy, Lead City University, Nigeria

ROQIB_AKINYEMI_Chapter_1_to_5 LCU LIBRARY

ORIGINALITY REPORT

14%

SIMILARITY INDEX

15%

INTERNET SOURCES

8%

PUBLICATIONS

3%

STUDENT PAPERS

PRIMARY SOURCES

1	matheo.uliege.be Internet Source	3%
2	www.thefreelibrary.com Internet Source	3%
3	www.cis.upenn.edu Internet Source	2%
4	www.scitepress.org Internet Source	1%
5	www.readkong.com Internet Source	1%
6	arxiv.org Internet Source	1%
7	web.archive.org Internet Source	1%
8	umpir.ump.edu.my Internet Source	1%
9	www.tandfonline.com Internet Source	1%

10 Mohammad Nuruzzaman, Omar Khadeer Hussain. "IntelliBot: A Dialogue-based chatbot for the insurance industry", Knowledge-Based Systems, 2020
Publication 1%

11 repositório.iscte-iul.pt
Internet Source 1%

Exclude quotes On
Exclude bibliography On

Exclude matches < 1%