FMDB ***Transactions on Sustainable Computing Systems***

AI Based Chatbot: An Approach of Utilizing on Customer Service Assistance

Rejwan Bin Sulaiman[1,*]

[1]Department of Computer Science and Technology, Northumbria University, Middlesex Street, London, United Kingdom.
rejwan.binsulaiman@gmail.com[1]

Abstract: The use of Artificial Intelligence (AI) and Machine Learning (ML) technologies in customer service has changed a lot in the last few years. The goal of this paper is to create an intelligent chatbot system that enhances company-customer interactions and improves the user experience. The chatbot serves as a virtual assistant, always available to answer client questions and provide logical, context-aware responses, utilising Natural Language Processing (NLP) and Information Retrieval (IR) methods. The research examines diverse chatbot designs, encompassing open- and closed-domain systems with retrieval- and generative-based responses, and evaluates their efficiency and constraints. The suggested system is mostly built on a closed-domain, retrieval-based design. This ensures that answers are accurate and reliable while reducing processing time. This paper also demonstrates how chatbots have evolved, from ELIZA to modern conversational agents like Siri and Alexa. It shows how important they are becoming for automated communication. The system's goal is to bridge the communication gaps that have long existed between businesses and their customers by utilising advanced NLP and machine learning techniques.

Keywords: Machine Learning; Retrieval-Based Response; Generative-Based Response; Information Retrieval; Statistical Machine Translation; Natural Language Processing.

Received on: 13/10/2024, **Revised on:** 07/01/2025, **Accepted on:** 01/02/2025, **Published on:** 05/06/2025

Journal Homepage: https://www.fmdbpub.com/user/journals/details/FTSCS

DOI: https://doi.org/10.69888/FTSCS.2025.000436

Cite as: R. B. Sulaiman, "AI Based Chatbot: An Approach of Utilizing on Customer Service Assistance," *FMDB Transactions on Sustainable Computing Systems*, vol. 3, no. 2, pp. 122-156, 2025.

1. Introduction

1.1. Overview

Recently, several firms have been adopting machine learning technologies to better serve their clients. This trend is common among companies that handle large, unstructured, and textual data. This includes those who specialise in marketing and business. According to research, the volume of unstructured data is growing exponentially [4]. Therefore, the primary concern is to have logical control of this kind of data, which is significant and requires development and improvement at all stages. Machine learning is a rapidly developing technology. A chatbot (a question-and-answer system) is one of the primary applications that uses machine learning. It has been a while; this system is in us. However, its operation is mostly based on factual Figures. There are techniques based on NLP and Information Retrieval (IR) that leverage statistical machine translation (SMT). NLP primarily focuses on providing answers based on datasets, and it has limitations in exploring the connections

[*]Corresponding author.

between words. However, IR focuses on generating new words and sentences based on previous replies. It is quite intelligent, as it has provided some logical answers by utilising machine learning techniques [5]. In this paper, we have developed a chatbot for the company that generates logical outputs using machine learning techniques. As the company requires an intelligent system to interact with its clients, we have focused on making the chatbot more efficient so that it can provide sensible responses to complex questions. This paper can be significant for many other firms that require an intelligent system to provide the best customer service. A chatbot system could be advantageous if its operation is integrated with social media and delivers prompt replies while handling a large number of products without human agents. This paper focuses primarily on the number of techniques employed in designing the chatbot system utilising machine learning technology.

1.2. Chatbot: Definition

In simple terms, a chatbot is a program that simulates conversations with a human user over the Internet. It is a machine-based, human-like agent available at all times to process enquiries. This chatbot system operates by understanding human inquiries (primarily in text form) and producing corresponding outputs. The history of chatbots dates back to the early days of computer science. It is understandable, as demonstrated by the simple test performed by Alan Turing, one of the experts in the late 1950s, which aimed to determine whether the person communicating with is a human or a computer without prior knowledge. This test has the great feature of making the system perfect in the sense that it becomes impossible to differentiate between humans and machines. In the real World, chatbot systems are still at an early stage of achieving that level of efficiency where it would be possible to chat about any topic predicted in the 1950s. This trend of chatbots understanding any topic can lead to a conversation flow that continues until the ultimate target is achieved. Experts and researchers have sought to improve efficiency by adopting numerous behavioural trends. For this paper, the consideration of a chatbot system is communication via text between the human and the computer program to process enquiries and provide logical output [6].

1.3. Chatbot: A Brief History

The origin of chatbots dates back to 1966, when a program called ELIZA was introduced, capable of rephrasing user input—a task now known as natural language processing (NLP). It was indeed one of the simplest chatbots, capable of processing user input, and it subsequently drew massive attention from researchers and scientists. Furthermore, this system is successfully able to fool many people. After ELIZA was introduced, the concept has been used for several decades, with minor changes and additions, including voice recognition and the understanding of emotions in various ways. Later in 2001, the new trend in chatbots, known as the Smart-Child, was introduced and operated in MSN and AOL messengers. By 2006, significant advancements were introduced in the chatbot system developed by IBM [3]. It was typically introduced to make it run to win one of the famous TV shows, Jeopardy. This strategy is supported by the application of an advanced NLP concept, which enables instant information retrieval. However, the major drawback was that it only followed up with a long cancellation, which meant it could not have a two-way conversation with any user [25]. Ultimately, by 2010, the most prominent concept of the chatbot was introduced, which is operated by virtual assistants like Cortana, Siri, and Alexa. The introduction of these virtual agents has brought a great revolution in the World of chatbot systems. These conversational agents engage in two-way conversations with human users, following a logical dialogue model. At the same time, Facebook introduced another significant improvement with Messenger, which can create conversational agents. As can be seen, significant development was carried out during the early phase of the chatbot system. The system is still developing and improving, even as many firms adopt this technology.

1.4. Motivation

Many researchers have been working in this field since the introduction of chatbots to improve the customer service experience (Figure 1). The following observations have been made after taking the survey into account:

- Most users are facing common issues with online communication channels. 34% say the website is harder to use (navigate), while in contrast, 31% report being unable to find answers to their simple questions on these communication platforms.
- On the other hand, users are also seeing some advantages of this system, i.e., 24/7 service (claimed by 64 % of the audience), quick reply (claimed by 55 %), and quick reply to simple questions (55 %).

The survey results above show that the expected outcomes are not achieved through this ordinary conversational channel. It shows that using a chatbot is not as effective as some alternative solutions. Currently, chatbot efficiency is still in its early stages, requiring significant improvements and further development. Around 53 % of the users from the survey claimed that the chatbot system is 'ineffective or 'rarely effective'. It shows the potential advantages of this technology, which can transform life, given the real-world implementation to date in this field.

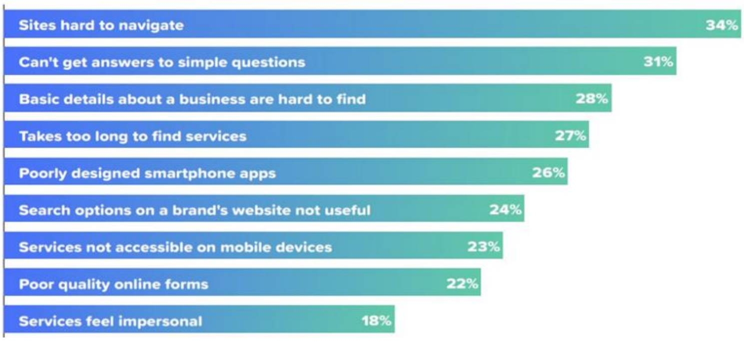

Figure 1: Customer service experience

The younger generation is more concerned about the positive impact of this system. After the survey, our primary motivation is to use this system for the potential benefit of the company. The use of advanced NLP techniques can certainly lead to improvement. As our company is primarily concerned with providing the best possible solution for this system to deliver the best experience for our clients, we are focusing on using it effectively and minimising the parameters that cause problems at the end-user's end.

1.5. Project Scope

This paper is designed for a company that wants to provide a 24-hour virtual agent to address clients' concerns and inquiries. The achievement of this paper is to be utilised by the company to implement it on its official website, making it available to clients and the public. As this is followed up with emerging technology, less information is available. The future impact of this paper is uncertain, as the system is new and under development, with ongoing improvements. The paper is designed based on recent research and available technologies; however, this will be updated and improved in the future for the best interests of the clients.

1.6. Hypothesis ANS Research Questions

The primary consideration for the company's chatbot system is to reduce. The chatbot is here to solve this issue. Implementing a chatbot system in the real world could improve customer service efficiency by making instant decisions based on human input. The proposed chatbot system for the company can significantly improve the customer experience by answering the questions, for instance, “What is the price of Heathrow from E6?”, “Where is my driver I have called for service?”. So, answers to these questions from a virtual agent can significantly reduce customer time and stress with instant replies. The primary focus of this paper is to enhance the customer experience with the company by implementing a chatbot system that is available 24/7 for assistance. In the implementation phase of the chatbot system, the following questions have been taken into account:

- How can the chatbot system be made more efficient for the company by reducing runtime?
- How can the chatbot system be user-friendly?
- How efficient and reliable is the chatbot's output in response to user input?
- How can a chatbot be considered an intelligent system for information processing?

While studying the above research questions, it is essential to understand the customer's experience with this new system. User acceptance, reliability, and intelligence are also factors we have looked into. Therefore, it operates based on how this system can benefit users, given the existing functionalized information available on the website. On the other hand, it is also necessary to perceive that the information provided by the chatbot system is valid and factual. The ultimate goal is to establish a system that can compensate for tcompensates

1.7. Aims and Objectives

The primary objective of this paper is to enhance customer experience by introducing a new method of communication between customers and the company. The objective is to develop a chatbot system on the company’s website that can effectively process customer enquiries. It includes even complex questions and provides several replies transparently. We aim to give replies to which sensible and logical answers can be given. The proposed chatbot system will utilise machine learning techniques to generate outcomes and retrieve answers from the database. In the first part of the paper, we have focused on the various research techniques available for the chatbot system. Afterwards, methods are selected that can process complex questions and provide definite answers.

1.8. Achievements

The following are some of the significant achievements of this paper that we aimed to gain:

- Design, develop, and implement a user-friendly chatbot that has been improved over time based on user feedback.
- Evaluation of this chatbot system, so that this system can be used and adopted by potential customers.
- Complete research work and learn more about the selection of the best techniques for natural language processing and other various tools that can be used in the implementation of the chatbot system.
- Learn in-depth about different machine learning techniques and algorithms.

2. Literature Review

2.1. Overview of Literature Review

The significant contribution to the company from my project placement involves the introduction of 'chatbot'functionality within the company's website, which is the cause of unprecedented change in the user experience. While implementing this system, we have reviewed the various techniques and tools previously used and still in use in the development of many systems. Upon analysing the literature, numerous techniques for implementing a chatbot are available, each differing based on various factors and functionalities [4]. For the development of this paper for the company, we have considered a variety of tools and technologies used in chatbot systems. For example, various NLP methods, Vector space models, and tools have been compared with a variety of machine learning models used in this system.

2.2. Chatbot System

In simple terms, a chatbot is a program that can engage in one-to-one conversations with users (humans) [4]. The program is specialised in efficiently interfacing with user inputs and, while using them, processes them using various methodologies (e.g., machine learning) to provide output to the user. The primary specialisation of a chatbot system is to ensure the user can have a meaningful conversation with another person (Figure 2).

Figure 2: General flow of chatbot system

The existence of this kind of system relies on various factors, including keyword matching, similarity in numeric or string data, and other natural language processing techniques.

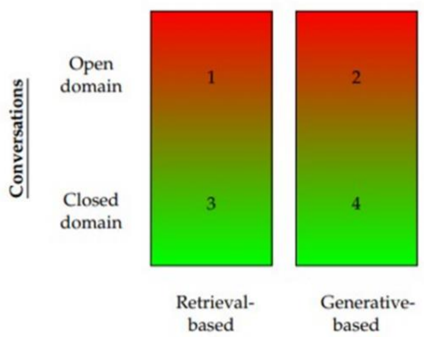

Figure 3: Types of chatbot based on conversation

Several chatbots have been implemented and are complex enough to understand user input. The chatbot is commonly used in many web applications that assist users by providing replies to their enquiries. Nowadays, there are several technologies and types of chatbots in use, differentiating in many ways, including functionality. The chatbot is available in various forms. Figure 3 above illustrates the primary categories of a chatbot. In contrast, an open domain is free to cover a variety of subjects, whereas a closed domain is restricted to specific areas. There is nothing very different from its operation. To understand this, consider Twitter status, which can be considered an open domain; ordering pizza online, by contrast, suggests a closed domain. Additionally, enquiries related to finance fall between open and closed domains [26]. Chatbots can be differentiated into retrieval or generative messaging. Retrieval is the process of using available data to obtain information. However, a retrieval-based chatbot follows a sequence of enhanced string and sentence matching and may also involve machine learning tools. These terms refer to the use of predefined techniques to produce an output from a given input. Generative chatbots are more complicated to generate responses than those using predefined material [3]. The following are different types of chatbots that are differentiated based on the responses they provide:

- **Open-Domain (Retrieval-Based Response):** These outputs, based on retrieval, rely on a fixed set of data, allowing any possible enquiry to be addressed. This sort of scenario cannot be implemented in the chatbot. Because the fixed set of retrievals is a hypothetical set that should cover any possible question a human could think of, obviously, that cannot be done, and hence, this type of chatting bot is not possible to create [26].
- **Open-Domain (Generic-Based Response):** It is also concerned with generating a valid response to any possible enquiry. The solution to these enquiries is based on the generic response, generally referred to as the AGI, an abbreviation for Artificial General Intelligence. It can be understood that the chatbot can perform similar tasks using its intellect, as humans do. Ongoing research is still underway in this sector [26].
- **Closed-Domain (Retrieval-Based Response):** This type of chatbot utilises individual datasets and text specific to the specified domains. Therefore, any enquiry generated is handled based on the output from those datasets. Nothing not available in the domain is not supported, as it does not provide solutions to inquiries that are not present in the domain dataset. So, while using this chatbot technique, most organisations do; in case of any enquiry not available in the domain, it is referred to a human for further assistance.
- **Closed-Domain (Generative-Based Response):** In this type of chatbot, an intelligent machine technique is used to provide a solution to the enquiry. The output is based on the datasets. However, it acts as a human, providing suggestions and a solution to the enquiry. The drawback of this kind of chatbot is its reliance on large training data, which exacerbates the problem and leads to significant grammatical errors in its outputs.

2.3. Comparison of Chatbots

While contrasting the various chatbots described above, several companies identify specific requirements in the chatbot selection process. Open-domain retrieval-based responses are not yet possible to implement. In various trials conducted by Google and Microsoft, no significant achievements have been made that would make this chatbot successful. Therefore, 'closed' is most often used in the selection process — for example, in financial domains [21]. Many firms commonly use retrieval-based and generative closed-domain models. However, work is still underway to improve the efficiency of these chatbots, and it certainly involves expanding their domains. The priority always leads to the best possible accuracy in grammar and other areas; therefore, to meet this requirement, a retrieval-based chatbot is an appropriate choice. It is predicted that the use of generative responses will be enhanced in specific ways by leveraging large datasets to obtain more complete answers [21].

2.4. Advanced Chatbot Systems

The chatbot is often considered a replacement for humans if it provides the best response. We can think of it as an online friend who is always available to have a chat. In simple terms, it is essentially a pattern-matching system, expertly designed to behave as closely as possible to a human. Nowadays, this match-patterning system is primarily used for automation and other chat channels as well. An automated online assistant (AOA) is a system that uses the same concept as a chatbot, allowing users to chat and perform simple tasks. Chatbots commonly utilise straightforward matching patterns based on the provided input; however, in contrast, AOA utilise more advanced methods, including NLP, named-entity recognition, and analysis, based on both semantic and non-semantic features. These techniques can also be used in question-and-answer; however, they are simpler, as they involve only answering the question rather than performing the task. The logic of QA can also explain this, as this system is only involved in answering questions, not the chat itself as a whole [6].

2.5. Functions of Chatbot

There is a broad spectrum of chatbot functionalities that are vital for many applications. The real need is to utilise this system to gather information from users and engage in conversations with them, rather than simply providing answers. The following are some of the methods of chatbots, which are based on their functionalities and limitations of their operation:

2.5.1. Enhancing the Chance of Right Answers

The fundamental requirement for companies adopting a chatbot is to enable their clients to benefit from it by providing accurate answers to enquiries. It is used to ensure the maximum probability of providing the correct answers (Figure 4).

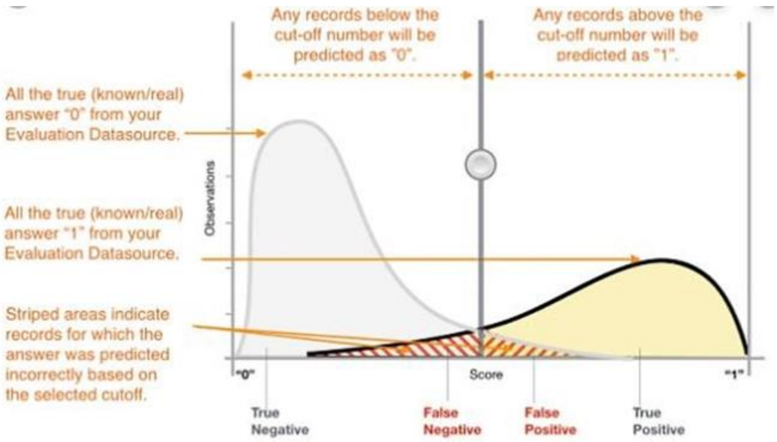

Figure 4: Binary classification of probability

- There is a variety of machine learning algorithms that can provide the exact probability of getting the correct answer for a given input. The following graph and Figures show the binary classification of probability in Amazon Machine Learning [7].
- So, while keeping this in mind, the chatbot can be designed to output an answer only if the probability (percentage) is high. However, if the certainty is lower, the program can send the enquiry to the expert, and from there, it reviews the completed form for all enquiries. Output follows up if the expert agrees with the answer, and it only requires a single click and send. In the event of any disagreement, it has the right to modify or format it.
- The simple method of chatbot operation involves a dataset that contains expert-provided answers and questions. Once a new question is received, it looks up a similar question in the dataset and returns the reply. For greater certainty, the chatbot follows up with a question: "Does your enquiry/question match with (dataset)?" In response to this question, if the input is positive, the chatbot will provide the correct answer; otherwise, it will refer the question to the expert for further processing [8].
- In the case of QA, there are always a variety of questions and answers; however, in reality, it follows up with similar questions (the same ones). In this method, it can easily follow up with a set of standardised questions and answers. Therefore, it can be classified as a question that can be entered in various ways. Some datasets are followed up with a higher likelihood of providing the correct answers to a set of questions of the same class [7].

2.6. Techniques of Answering

There are various techniques in use, depending on the AI system. Information retrieval is an approach used to solve queries. It is originally based on accessing data and is used to retrieve a list of possible approaches to the enquiry rather than just answering. Hence, it provides the user with access to the information, enabling them to track it and find the exact solution to their inquiries. This can be easily understood with the example of the Google search engine. It follows up with the user to ask a question in simple words and retrieves a list of short answers to the question [9]. In our case, it is essential to give a specific answer; it might include a link to other websites; however, it will not retrieve information based on the list of documents. Technically, we presented the best answer based on the highest probability of certainty, followed by a list of answers to the questions.

Question Reply: Sometimes, an additional question from the user is necessary to follow up on a complex question that requires more detailed information to process. e.g., if a user asks when he is eligible for a pension, the system will regenerate a question asking for that person's date of birth. An excellent feature of the chatbot system is that, when replying to a new question, it helps retrieve the exact information about the user, ultimately yielding the correct output. It is not easy to generate a follow-up question while maintaining the context of the enquiry, as this requires the client to provide the precise information needed and to inform them what information is missing. To implement this design in a chatbot system, it is necessary to use a template that facilitates information retrieval. If the question is linked to the most frequently used questions, the program will look up the year of birth mentioned. If it is not there, the program will ask for the date of birth. This kind of system doesn't.To provide flexibility, it can be used only in limited, specified cases. Currently, no technology has been developed to solve this in any alternative way [10].

Natural Language Processing (NLP): It is a process that enables machines to learn and understand human language. It utilises a hierarchical procedure for processing natural language. This kind of language processing is considered complicated in the IT world, due to the complexity and ambiguity of natural language. This system operates on the principle that, if a computer needs to comprehend the meaning of natural language, it does not involve only the meaning of individual words or sentences; rather, it requires understanding the entire concept. NLP is divided into the following components [11].

Morphological Analysis: This type of analysis involves understanding the smaller elements that comprise a word. In the first level of natural language processing, it only looks up for the words that are taken from the text, and at this stage, it does not contemplate the punctuation of the whole text. Moreover, these are further narrowed and analysed into smaller components. The following figure illustrates the algorithm used in morphological analysis (Figure 5).

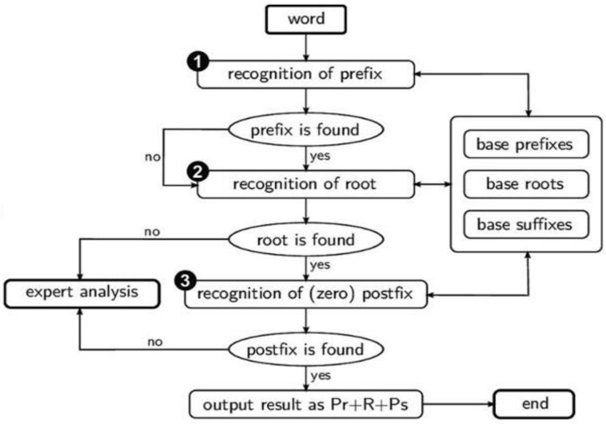

Figure 5: Morphological flow

The following are some of the essential terms used in morphological analysis:

- **Tokenisation:** Techniques for breaking text into words, sentences, and other smaller components are referred to as tokenisation. Those tokens that are processed are used as input for further processing. Furthermore, at this stage, a tokeniser is required to follow up on the removal or transformation of the abbreviation, punctuation, and other white spaces. A significant aspect of the tokenisation process is identifying useful or meaningful words across the entire text.
- **Removal of Stop Words:** These are commonly used words in natural language, i.e., a, 'of', 'the', etc. Their words are not used to separate the context; instead, they are used to join the sentences. If these words are not considered essential in the text, they can be removed as per the domain. Removal of these words is quite a difficult task, as it involves distinguishing between meaningful and meaningless occurrences; therefore, generating a stop-word list is not easy. It is essential to remove these stop words, as this will reduce data size and ultimately improve classification performance across the whole text [24].
- **Stemming Process:** This process is used to reduce words to their base form, also known as their stem. Moreover, it causes the words to come back to their original form. It can be understood by considering the words "speak", "spoke", and "spoken"; these three words are classified as one word, which is "spoken". This method makes it easier to match the text to the same context.
- **Auto-Query Expansion:** This involves adding words to the original query, which improves its matching within the text. Some conventional methods for auto-query expansion include synonym expansion, stemming, and spelling correction.
- **Part of Speech (POS) Tag:** It is the process of tagging words in a text according to their PoS. This is illustrated by the example of tagging a word with its part of speech (PoS). This process can be done using either a rule-based or a statistical approach. Using this PoS-based tagging is quite useful for generating output by parsing the sentence, thereby reducing ambiguity in NLP.
- **Syntactic Analysis:** It is a type of analysis that focuses on the components of sentences. The second stage of NLP explains how words are transformed into structures, which can show how sequences can be related to one another. Some terms are vital for syntactic analysis. These are discussed in the following.
- **Parsing:** It is an operation that transforms the sequence of words into a formal structure based on grammar. Parsing produces a tree that explains the relationship between the input and the words in computational linguistics. Deep parsing

is responsible for the complete tree building, whereas shallow parsing only builds a portion of the tree for a single sentence [12]; [1].

- **Bag-of-Words:** This model represents the texts straightforwardly. A text is displayed like a group of words where the words have no relation to each other. Besides, no grammar is acquainted with the texts. This model can only display words and their multiplicities and is used for document classification.
- **N-Grams:** The N-gram model is widely used for storing spatial information, whereas the bag-of-words model captures only a small portion of the text. It has been stated that this model is widely utilised to stem. Using the model, it is possible to anticipate the use of a word in a sequence. The total number of words that have been regarded can be denoted by N in this model [13]. For example, if N=2, the sentence transformation may look like, "Where do you want to go?" into different segments like "Where do", "do you", "you want", "want to", "to go". This anticipation of the recurrence of particular words can result from the frequency with which the words occur. However, in practice, it is unlikely that the word combination will appear in the dataset. This problem can be solved by using the probability distribution of the words that appear [14].
- **Named Entity Recognition (NER):** NER is responsible for accurately identifying and labelling people and places. A significant amount of research is being conducted in this area. The current approach of this report is to trigger and identify a pattern that matches an option for lookup. In the case of a lookup, the system can only recognise the stored list. Also, it is a fast approach that can be applied to other texts. However, there are fixed costs for maintaining and collecting the entities, which create definite problems with name variations. For instance, "Mount Everest "can be considered, which can be recognised as "Mount + Capitalised Word". In pattern matching, patterns are constructed manually. For instance, "<Name> lives in <Location> on <Date>"will yield the result of "Jessy Lives in Boston for 3 years" [2].
- **Word Sense Disambiguation:** This is used to determine the meanings of words used in texts. It is known that a single word can change meaning, and it is one of the difficult tasks to differentiate between words. The associated methods for disambiguation can include knowledge-based, empirical, and AI methods [16].
- **Latent Semantic Analysis (LSA):** LSA is a methodological process that assumes words with identical meanings will most likely appear in the same texts. The terms and documents outlined in this LSA are analysed. To contain the counts of the words, a word-document is formulated where the text documents are kept in columns, and the unique words are kept in rows. Similarities are preserved between the texts, and the reduction of unique words is achieved using singular value decomposition [27]; [28].
- **Semantic (Role) Labelling:** Semantic labelling identifies the actions of words in texts. An example can be given in this regard, such as "Jessy lives in Boston". The semantic labelling will be able to identify "Jessy" as the person who lives, the verb "lives in", and "Boston, as the person's habitat. It enables the reader to understand the meaning.
- **Lexical Answer Type (LAT):** LAT is responsible for indicating the entity [23]. An example can be given to better understand it, such as "Who is the president of the USA?" Here, LAT is the "Who," since it identifies the person responsible.
- **Relation Detection:** Relation detection is used to discover relationships between syntactic and semantic elements in sentences [23]. An example can be given, like, "Who is the president of the USA?", the relation can be found in the ad (president, x, the USA)

2.7. Vector Space Model

The vector space model was initially utilised in the SMART Information Retrieval System [30]. This mathematical model can be used to determine the vector representations of textual documents. The comparison can be made using this model; hence, the calculation can be performed to identify the same items between the documents. This vector space model creates a matrix and then assigns values, which are compared after the words appear in text documents. One of the most famous weighting systems is TF-IDF, which was already explained in full in the prior section of this paper report. The total number of words is the primary factor in determining the dimensionality of the vector, such as the total number of words in the documents and the words being compared [26]. To compare the text documents, it is necessary to perform vector operations. Operations are measures used to find similarities. The similarity measure operation can provide a ranking that depicts relevance, and then a comparison will take place between the text documents. The similarity can be measured using a formula known as 'cosine similarity'. The formula is given below:

$$\cos\theta = \frac{\mathrm{d1}\,.\,\mathrm{d2}}{\left|\left|\mathrm{d1}\right|\right|.\left|\left|\mathrm{d2}\right|\right|}$$

Concerning the mentioned formula, d1 ·d2 denotes the intersection of two-document vectors, whereas ‖d1‖ ·‖d2‖ denotes the length of the two document vectors to have normalisation in the scorecard. The higher the cosine value, the greater the similarity between the documents. It is imperative to normalise, as it can provide greater confidence that a few documents will match a longer version. One limitation of the model is that it is expected to produce a word-for-word match with another document. However, by utilising various synonyms and pre-processing techniques, this problem can be avoided [18].

2.8. Machine Learning Models

Machine learning is a field of computer science in which computers learn without being programmed to do so. Machine learning algorithms enable computers to make data-driven predictions efficiently, eliminating the need for rigorous programming. Different machine learning methods can be classified in numerous ways. Supervised and unsupervised learning are two methods of classification. In supervised learning, the computer, along with the anticipated variables, can be presented with a view to their use. This learning intends to create a pattern that aligns with the anticipated values in the results. On the other hand, unsupervised learning lacks labels, meaning it has no variables. A dataset can be observed with different observations across numerous sets of variables. The algorithm is responsible for the data structure. This type of learning structure can be used to discover hidden patterns in the data [19]. The type of total output generated by machine learning can be classified. The classification can provide the answers in multiple classes. The model has been used in a supervised learning setting, which indicates the new class to which it belongs. The regression method can provide supervised solutions but cannot produce categorical outputs, as it produces only continuous outputs. A clustering method can classify aggregate data even without prior knowledge of the groups. This section discusses machine learning techniques for chatbots. Also, emerging techniques will be discussed here. Though there are many techniques in machine learning, question-answering systems have been developed in this context [19].

2.8.1. K-Means Clustering

K-means classifies n observations into k groups. The notion of this idea promotes the observation that belongs to the closest mean [24]. The algorithm has functionality that can be divided into five categories:

- **Choose k:** The number of clusters.
- Initialisation of centroids.
- Assignment of points to the cluster, along with the centroids.
- Re-arrangement of centroids based on the formulated clusters.
- Re-assignment of points to the cluster along with the centroids.

The last two steps should be repeated until the data points do not have a fixed position. Additionally, the data points should have fixed positions and should not be assigned to the nearest cluster. The initialisation of the centroids can be done at random. However, it is essential to utilise the heuristic located at the farthest point. In this case, though the first centroid was selected randomly, the second centroid was selected from the farthest point [20].

2.8.2. K-Nearest Neighbours (KNN)

Classification and regression can be performed using the supervised k-Nearest Neighbours (kNN) algorithm. In terms of classification, a new point can be assigned a label k. The assignment of the neighbours to the k is a significant vote in classifying the new point. Here, the classification is into two classes, each with a different colour, such as class 1 (green) or class 2 (blue). If k=3, it depicts the nearest neighbours to the red star; it would then be classified as blue (class 2). However, if the attention can be placed in k=5, then the star will be given to green or class 1. This provides information on kNN's sensitivity to the structure (Figure 6).

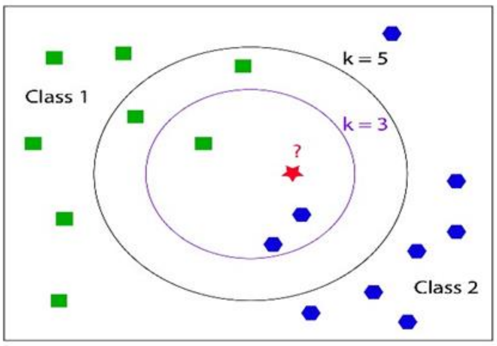

Figure 6: KNN classification algorithm

In the case of regression with the kNN input, k is the number of neighbours, whereas the overall output is a property value. The property value is taken to be the mean of all the values. In both regression and classification with KNN, the overall algorithm can be improved if the distances to the neighbours are computed during the process. Afterwards, the assignment of weights should be done to the nearer and farther neighbours. More weight should be given to the close ones, and less weight should be given to distant ones. Here, d signifies the distance between the neighbour and the new point [21].

2.8.3. Random Forest

One of the widely used machine learning models is the Random Forest, which can perform both regression and classification. It combines several weak models into a large, robust model. The smaller models are represented as decision trees, splitting every datum at every possible node. Several decision trees can be formulated upon utilising Random Forest and its unique conditions, as well as features on those nodes. It is supposed to have different nodes in each tree at the time of selection. Only a handful of all the features can be provided as regular options. To appoint a reasonably new phenomenon to the class, each tree is required to provide a classification, which is known as a vote in some cases. This model selects the class with the most votes across all trees [29].

2.8.4. XGBoost

XGBoost, also known as eXtreme Gradient Boosting, is an emerging gradient boosting method that has gained popularity among professionals due to its significantly faster execution compared to the Instant Gradient Boosting model [34]; [32]. Numerous weak rules are combined via boosting, and voting is also used to build a powerful model, such as the Random Forest. An example is a weak rule: "If there are questions that have five identical words, then they would be identical to each other too." The rules can be discovered using machine learning algorithms [27]. XGBoost is unique for its speed, which is enabled by parallel computing. Additionally, various types of benefits can effectively mitigate overfitting. Additionally, this provides the user with flexibility, as they can evaluate different criteria and set the optimisation objective [21].

2.8.5. Support Vector Machines (SVM)

SVM is a binary classifier trained on independent variables. It is a type of learning method that can produce desired outputs. The boundary that an SVM can create is called the maximum-margin hyperplane. However, to be able to discover the hyperplane, it is imperative to maximise the margin by increasing the perpendicular distance between the observations of different classes. In Figure 7, a hyperplane is depicted with different types of small black lines [22]; [24]. The bold ones are the exact maximum-margin hyperplanes, since they can create a much larger margin than the hyperplanes for green or orange ones. When adding a new dataset, it is imperative to know the location of the maximum-margin hyperplane.

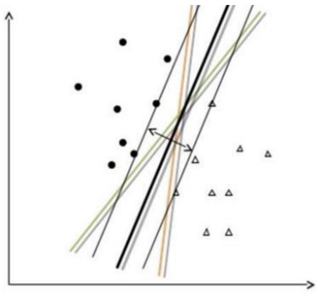

Figure 7: SVM hyperplane margin

When using a linear hyperplane, I can become confused in some cases when classifying data into two distinct categories, as shown in Figure 8 below.

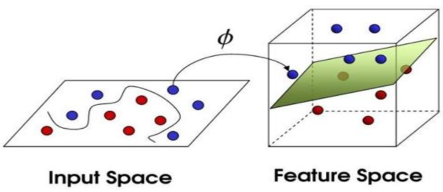

Figure 8: Non-linear SVM using kernel [27]; [33]

In a higher phase or dimension, this problem can be solved, provided the SVM is taken into consideration. The kernel function is responsible for calculating higher-dimensional quantities [23]; [24]. When it is time to translate the linear boundary, the input space turns into a nonlinear space.

2.8.6. Neural Networks

The artificial neural network is created on the structure of the human brain. Neurons in the human brain are interconnected, forming cells that enable humans to work and think more efficiently. Figure 9 below illustrates a network known as an 'Artificial neural network'. In the human brain, a network processes information and delivers it for further use. Signals can pass through neurons and form connections. Every neuron performs a specific function, so its output is determined by the inputs it receives. Hidden layers are present in an artificial neural network, and the number of layers can be large, exceeding the number shown in the depicted Figure. The input layers can have numerous neurons and serve as independent variables, whereas the output-layer neurons can serve as dependent variables. However, the total number of hidden layers, as well as the number of hidden layers, can be calculated only from the number and types of hidden layers.

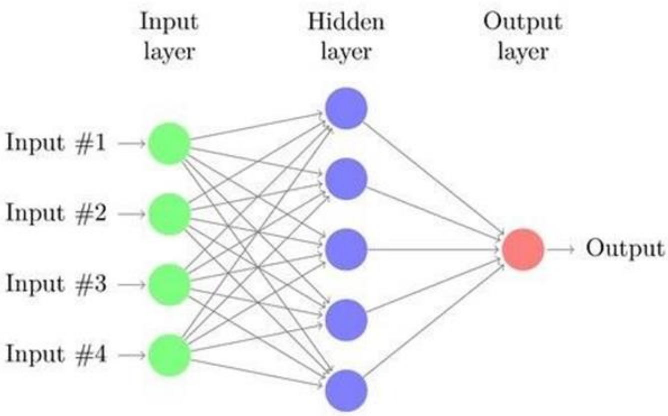

Figure 9: Artificial neural network layer

The neural network can be trained on the linkages among different types of neurons to determine the optimal model for the training dataset. Work certainly needs to be done with feedback, rest assured, and the network should be capable of understanding right and wrong. With all the necessary information available, the modification should be made to the connection to verify whether the result is satisfactory. The alteration to the connection will put weight on the difference between the actual income outcome and the model outcome, which differs significantly. The entire process is known as backpropagation in the context of neural networks. The functionality of this network begins with the output neuron and proceeds through hidden neurons, which are designed to interact with the input neurons. New responses can be created after the network is trained [33]. Numerous types of artificial neural networks are available. Figure 9 above shows the Feedforward Neural Network (FNN), which allows information to flow in only one direction. In chatbot systems, Recurrent Neural Networks (RNNs) are widely used because they can learn broader contexts, making them a unique type among neural network models. RNNs possess bidirectional data propagation, allowing information to flow in both directions from input to output [24].

2.9. Critical Evaluation

As discussed in the literature review above, various tools and technologies can be adopted to develop the chatbot system for the website. The selection of specific tools and technologies follows the company's criteria, and, of course, the suggestion has been proposed by the IT expert, ensuring that the developed system is sustainable and promising. The technologies mentioned above provide an overview of possible solutions for implementing the system; however, in this section, we will carefully analyse the reasons for selecting the tools and techniques adopted in our paper. As mentioned earlier, Natural language processing (NLP) technology utilises specific tools and techniques that enable it to understand human language, making it quite user-friendly for issuing clear commands to the machine in a natural way. This set of rules does not apply to several word-processors. The uniqueness of NLP technology can be measured by understanding the hierarchical structure of natural language, i.e., letters are combined into words, and words are combined into sentences. However, from another perspective, NLP is not asimple as it looks; in hais ambiguousnd it The justification lies in the fact that, when a machine (computer) which will be running NLP is required to know/understand not just the meanings of the word or text, it will also have to believe the actual concept of the whole sentence/words to output accordingly [31].

Nevertheless, the vector space model can address this drawback by treating words as vectors, based on mathematical principles. This vector-based model can be used to find the similarity between two documents and follows the calculations. Initially, this model was beneficial for data retrieval. Thus, it was considered the most efficient model for retrieving information from the database. The vector space model utilises a quite clever technique based on the leading of the document-term matrix, which allocates an absolute value to the term only when it is required for comparison. To perform this action, there are multiple ways, e.g., term weighting. However, the most common approach is to follow up with TF-IDF weighting, as discussed in the previous section. The exact vector dimensions depend on the total number of words used to compare multiple documents [29]. To compare the various vector-based text documents, several operations are used. In other words, these kinds of vector operations are known as similar measures.

This operating system provides ranking criteria based on the relevance of the documents when comparing and contrasting them. However, the most commonly used similarity measure is known as the cosine measure. Moreover, as mentioned above, the models do not provide us with language independence. For our company, we need such a technique that is language-independent and can converse in any language at any time. Nevertheless, with all this, there is a unique method that covers several drawbacks in other technologies, as mentioned above, i.e., machine learning technology. Various machine learning techniques can be used, each quite useful in its own way. One of the most popular ways is to follow a learning technique; now the concept of differentiation is based on the type of learning technique, i.e., supervised or unsupervised. The supervised learning process is followed by the presentation of specific datasets, consisting of predictive variables and the results they produce. The basic concept of supervised learning is to learn a rule or follow specific patterns or combinations, such that it can match the results produced by the variables. However, in contrast, unsupervised learning does not yield predictive variables or outputs. It consists of specific datasets comprising the absolute value, followed by the variables utilised for observation purposes. Moreover, it ultimately follows algorithms to determine the order of the data. It is one of the prominent types of learning that can uncover the hidden structure and order in data. While considering the differences mentioned above, we have carefully analysed the various machine learning techniques. Moreover, after taking this into account, we have selected a supervised learning technique for our paper.

2.9.1. Operation of Chatterbot

The machine learning technology we have adopted for our paper utilises a chatbot. The chatterbot uses a Python library to generate responses based on user input, providing automated responses accordingly. The operation of the Chatterbot system is entirely based on the machine learning algorithms that can generate a variety of responses. This type of behaviour creates a friendly environment for developers to utilise the technology in the chatbot system to generate automated responses based on user input [35]. The processing of a chatterbot uses a variety of machine learning algorithms. The selection of an algorithm depends on how the chatbot is intended to be used and its configurations. The following are some conventional machine learning techniques used in the chatterbot system.

2.9.2. Searching Algorithm

Searching is a prominent component of artificial intelligence. There is some technical differentiation between machine learning technology and artificial intelligence. Using the search algorithm is crucial, as it is the backbone of the chatbot system, responsible for performing instant actions based on the user's input. The following are some of the issues of the chatbot system to generate a response based on the user input:

- The system already knows how to match input statements.
- The number of times a matching response is generated.
- The probability of input statements being known by the system [36].

2.9.3. Classification Algorithm

Various logic adapters are accompanied by a chatterbot that utilises Bayesian classification to determine the matching pattern of input statements, which are regulated by a specific set of rules for generating responses. In machine learning, a Bayes classifier is a probabilistic classifier that uses Bayes' theorem. It is considered one of the simplest Bayesian network models. To understand the nature of the problem or learning purpose, the Bayes classifier is considered a scalable approach. The real operation of the Chatterbot system is initiated when an untrained instance of the Chatterbot is started, with no prior knowledge of its internal workings. Whenever a text or statement is entered into the system, the steps are to save the input in the library, process it, and produce the Responses accordingly. The operation of ChatterBot improves, or, in other words, the accuracy of its responses increases as the number of received inputs (i.e., statements or texts) increases [27]. In simple terms, the response is based on the amount of input received, as it can generate responses towards the provider when new input is received. When the program is specialised, it is specialised to the best possible extent. In return programs, the closest-matched statement generated earlier is used, along with the best response generated on that occasion. Therefore, the program carefully analysed known input responses and, based on that, when a new input is received, it considers all known responses to similar inputs and thus produces the best possible response, thereby improving the system's overall performance. The following are some of the reasons for choosing the machine learning technique over the three other technologies mentioned above:

- It makes the system (i.e., the chatbot) very user-friendly; even a non-technical person can be trained to operate it using machine learning techniques, whereas other technologies require excellent technical skills and qualifications.
- The use of machine learning can ease conversation updates, enabling it to take instant input, process it, and provide the corresponding output.
- It also offers the benefit of ease of adding or deleting topics, unlike the other mentioned technologies.
- It allows adding multiple languages, even those that are very different from English, such as Arabic, Hebrew, and Bengali. This differentiation makes the system more reliable, as the chatbot can be built for any language, topic, and institution. ChatterBot's language independence allows users to talk in any language of their choice. As the chatterbot operated using machine learning, the responses generated can be used to enhance the likelihood of responses based on specific knowledge, along with human interactions and gathered information.
- One of the best advantages of studying this machine learning technology is that it enables the addition of any topic of choice without restrictions. However, this is not the case with other technologies.
- Machine learning technology can also interact with websites in a friendly manner, whereas the other mentioned technologies require more complex interfacing.
- Furthermore, the primary concern with this choice is that utilising this technology ensures instant, delay-free system operation.
- The accuracy of this technology is much higher than that of the other technologies mentioned above.

For the machine learning technology, we have carefully analysed various concepts used. For the implementation of our Chatbot project for the company, we have used the Chatterbot concept, one of the best Python libraries for machine learning.

3. Methodology

3.1. Overview of Methodology

In this section, we will outline the design and development of a chatbot, as well as the data collection method. We will also explain different SDLC models and mention which models we have chosen to accomplish the paper, with the phases of chatbot development. The chatbot is entitled to formulate a response after receiving a user's question. Additionally, users can engage in a conversation with the chatbot. The methodology for development and execution will be described in full in the following section, which accompanies the machine learning model.

3.1.1. Artefact Development Methodology

Different methods can be considered to complete the paper [38]. In the case of web- and software-based projects, some models can serve researchers well, offering significant benefits by having the potential and capacity to address the paper's objective. The models are:

- Waterfall Model
- Spiral Method (SDM)
- Iterative and Incremental Method
- V-Shaped Model
- Evolutionary Prototyping Model
- Agile development

3.1.2. Waterfall Model

Upon giving a thought, it has been concluded that for this paper, the waterfall method can possess significant benefits in the paper making, and the benefits of the waterfall model are given as follows:

- Structured and definite phases.
- Orderly management.
- Time management from the initiation to the completion point.
- Unique deliverables in the System Development Life Cycle (SDLC).

- Using the sequential model, each phase can be visualised as a downward flow until the end of that phase [37].

Figure 10 below illustrates the system development lifecycle, which commences with requirements analysis. Gradually and sequentially, design, implementation, testing, deployment, and maintenance take place [38]. It is essential to ensure that each phase is completed before the next phase can begin. Only one phase can proceed, while the others should remain unchanged.

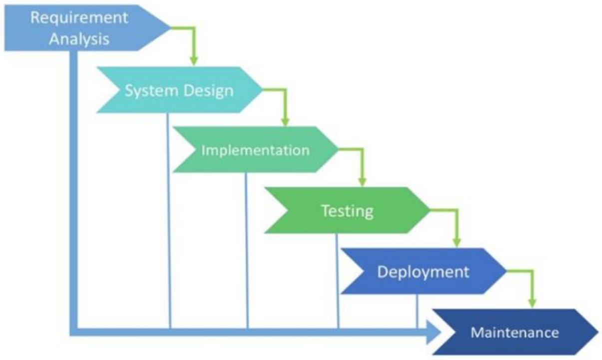

Figure 10: System development life cycle, waterfall model

3.1.3. Advantages and Disadvantages of the Waterfall Model

If the paper definition is stable, the requirements can be documented precisely and accurately. One of the model's critical features is the predefined technology stack, which makes it delicate. Also, there are no ambiguous requirements for the model if the assigned paper is short.

Advantages	Disadvantages
The usefulness of the Waterfall SDLC model lies in its ability to be utilised quickly and efficiently, making it a widely used SDLC model among researchers.	One disadvantage for researchers who adopt this model is its readiness. This model is not ready for practical use until the last stage is completed. If any work remains to be completed, the model fails to perform as intended.
Researchers need to know the results for each phase so they can become well familiar with the likely outputs they will produce. This model enables researchers to view the output after each phase. Additionally, a review of the process is conducted after the phase has been completed.	This model exhibits high uncertainty and poses a risk to the researcher due to its lack of critical functionality. This model has limited functionality within which it works well, and cannot push boundaries to work smarter.
The development stage starts after one stage is finished. In this way, the researchers can review all the views before initiating the next stage.	This model is beneficial for short, straightforward papers. However, this model is not particularly useful for complex papers and struggles to handle complexity efficiently.
This model is suitable for papers with clear instructions and fewer functionalities and complexities. Thus, it is better to use this model when the paper is not very large.	It is not recommended to use the model if the paper requires numerous functionalities and complex work capabilities. Hence, small and medium-sized papers can adopt the model, whereas large projects should not.
During the model's development cycle, it can identify its critical points.	On another note, it isn't easy to measure progress while the model is in development.
Classification and prioritisation of tasks can be performed within the model, as it is segregated into different folds.	One of the major shortcomings of the model is its inability to identify and rectify problems in advance. Hence, the integration is done at the end of the model, which takes time and effort for researchers to implement.

3.2. Data Collection Requirements

A data collection methodology needs to be formulated, as the data collection can differ across cases. Data can be collected through qualitative, quantitative, or a combination of both types of analysis. However, this paper is based on a product. It is essential to incorporate both qualitative analyses from the literature review and quantitative assessments of customers' perceptions of the product. Hence, data collection methodology will involve the following methods to collect data from the respondents.

3.2.1. Interview

It is a data collection methodology in which researchers prepare specific questions to ask respondents, who then answer them. In this method, data collection may take time, but the collected data is authentic, and respondents are less likely to provide inaccurate information when they are not being probed [15]. In our research, we have conducted regular interviews with our existing customers to address their queries, as well as with other customers who have similar needs. Based on the interviews, we have built the chatbot training module and trained the chatbot.

3.2.2. Survey

The survey data collection process is considered the backbone of primary research. It is a data-collection method that requires researchers to combine diverse research interests into a single study, enabling them to gather data on different bases from the same respondents. So, collecting a large pool of data may be possible upon embracing this procedure. Therefore, this paper can be served by the offerings of this data collection procedure and will also benefit from them. Over the paper, we have taken multiple approaches to collect data. We have focused on collecting data through surveys rather than other methods. The survey questions are analysed in the thesis's analysis section.

3.3. Software Methodology

To complete this paper, numerous software methodologies have been studied. There are ample software tools available for this purpose. However, not all software tools are suitable for this paper's needs. After evaluating numerous software tools, PyCharm has been considered an IDE for Oracle's development kit due to its benefits. To achieve a fast and reliable application, PyCharm is one of the most widely used software tools that support researchers and practitioners in that field. Different IDEs, such as Wing IDE, Eclipse, and PyCharm, could have been chosen for the paper. Since improving efficiency is a top priority for this paper, PyCharm has been chosen to ensure reliability and speed, which are not guaranteed in other IDEs.

- **MySQL DB:** We have given MySQL DB higher priority than other SQL server-based databases due to its scalability, performance, reliability, and ease of use. It is now the world's most popular database.
- **Sublime Text 2:** We have used the Sublime Text editor, overriding other popular text editors. There is a reason we chose Sublime Text over others. Reasons are stability, customisability, featurefulness, and cross-platform support.
- **PhpMyAdmin** is a free and open-source tool for administrative purposes in web development.
- **Cpanel:** Multiple features that can be benefited from if cPanel is used in the hosting service, like creating/deleting / forwarding / automatic spam control for email accounts. Moreover, cloning a website, installing any app in the native language, file restoration, and backing up. It is the featured options that we cannot get through other service providers.
- **Oracle VM VirtualBox:** It is a free and open-source tool for experimenting with different hardware configurations that might cause issues with the host computer.
- **Nginx Server:** We have chosen Nginx because it is the best on the market, offering advanced features such as media streaming and HTTP reverse proxying. On the other hand, it also delivers high performance for dynamic websites.

3.4. Hardware Methodology

If one focuses on the methodological aspect of the hardware, it is evident that this paper is not based on the core of the hardware (Figure 11).

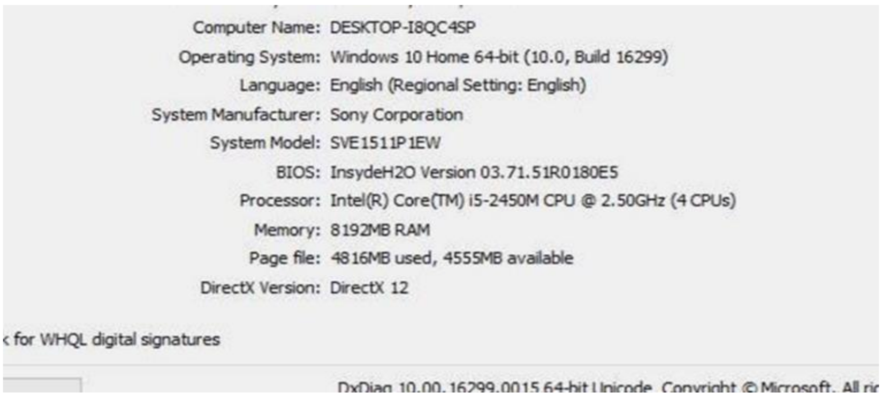

Figure 11: Configuration of the computer

3.5. Functional Requirements

- Permission to use the application for unregistered users.
- Assisting users to satisfy their queries.
- Conversing with the users through text commands.
- Understanding the users through natural language.
- Staying in a conversational state, even if the chatbot may not fully understand what the user says, it should request more detailed information.

3.6. Non-Functional Requirements

- Efficient in responding to fast messages.
- It should take no more than 5 seconds to reply to users.
- Should be bug-free and reliable to the users.
- A scalable database to provide services to a large number of users.
- It should be secured due to the sensitive information it contains from users.
- A two-factor authentication system should be introduced to ensure robust security. We have not implemented this feature yet, but it could be done in the future.
- Should have compliance with the GDPR, which is a form of data protection law.
- Promotion of human-computer interaction using natural language as a medium of communication.
- Providing accurate and definitive responses to users' queries.
- Taking care of unexpected inputs by providing corrective measures to users, assisting them efficiently and effectively.

4. Analysis

4.1. Overview of Analysis

In our case of data collection, we have adopted two ways, i.e., interviews and surveys. For this data collection methodology, we have selected a few people to help us with surveys and interviews. The group we have chosen consists of individuals aged 19 to 63, totalling 25 members. We conducted this survey using SurveyMonkey on 05th May 2019. The following are the questions and the responses from the participants

4.1.1. What Means Have You Taken for Business Communication in the Last 12 Months?

The following bar graph displays the response from the participants (Figure 12).

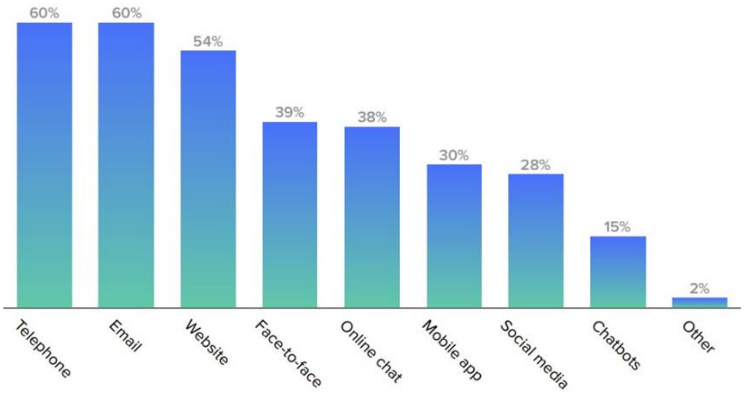

Figure 12: Different means of communication

The result from the survey question above shows that 15% of participants have used a chatbot system to communicate with the business over the given period, i.e., 12 months. However, it can also be seen that around one-quarter of people have used telephones and email for business communication; that's quite a significant amount. We can predict an increase in the number of users in this category over the coming years. Based on the analysis above, we tried to identify the reasons why people are not considering the chatbot as a medium for business communication. A chatbot is an emerging technology that is still in its early stages of development and efficiency, providing a better customer experience when answering their queries. To gain a better understanding, we advised our participants to consider other online services they use daily, such as search engines, websites, and other apps.

4.1.2. What Was Your Worst Experience While Using Online services over the Last Month?

According to the bar graph above, the worst experience for around 34% of participants is their struggle with site navigation. Around 31% reported not receiving solutions to their answers, and around 28% reported that the necessary business details are harder to find. Based on the above results, it can be concluded that people's experience with online business communication is generally poor. Users prefer information from the support team, indicating that the online system is not meeting their expectations efficiently. Hence, users are looking for instant, high-quality services (Figure 13).

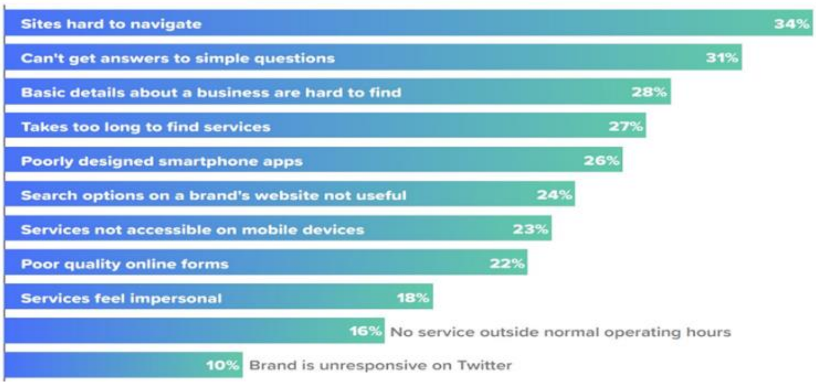

Figure 13: Online experience survey result

When companies fail to provide the expected information on their online platforms, users often switch to competitors in the hope of receiving better service. Therefore, this user drives the need for an efficient, instant, and high-quality solution, and that's where the concept of using machine learning in chatbots comes in. While continuing the survey with the same participants, we have asked the participants the following question to analyse the importance of the chatbot system:

4.1.3. Why Do You Think Chatbot Systems Should Be Used in The Future?

The bar graph displays the results from the survey mentioned above. It can be seen that the highest number of participants (i.e., 37%) prefer chatbots in the future to obtain instant solutions to their queries. However, 35% of participants also think it could be used to resolve issues and complaints. Interestingly, 34% of participants use the chatbot solely to connect with the human service assistant (Figure 14).

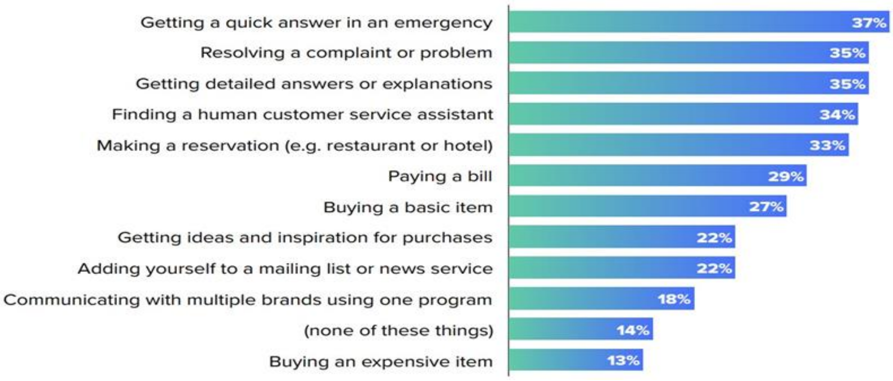

Figure 14: Result of the survey on the future of chatbots

These results indicate that users are primarily concerned about the chatbot system's ability to replay messages instantly and provide general information. However, for complex queries or purchases, most users prefer to connect with a human assistant. While contemplating the above assumptions and results, we continued the survey by asking the following question to gain a better understanding of the chatbot system from the user’s perspective. Let us find out whether the chatbot system is working efficiently on the online platforms you use regularly.

4.1.4. Which one of The Following Advantages Would You Consider for The Chatbot System?

The bar graph above displays the results for the counting chatbot used in online services. The results show that most participants (i.e., 64%) will consider the chatbot's advantage of 24-hour availability (Figure 15).

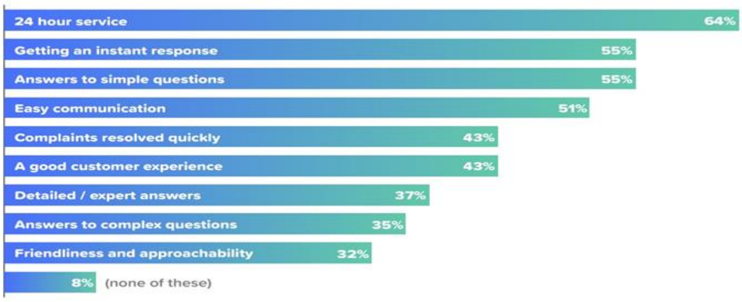

Figure 15: Advantages of a chatbot system

In contrast, around 55% of participants consider the benefit of this technology in generating instant responses, and the same participants believe it can be quite useful for providing simple answers to questions. We have continued asking for more expected experience from the participants by asking the following question.

4.1.5. Why Would You Not Consider a Chatbot System and What are the Advantages and Challenges of Doing Business Communication?

The above bar graph illustrates the results of not considering the chatbot system, showing that the largest number of participants (i.e., 43%) prefer speaking to a real person rather than relying on technology for their concerns. Thirty per cent of participants are not confident because they think it can lead to mistakes. Some say (27%) that a chatbot should be considered only for social media, i.e., Facebook. One notable finding was that 15% of participants stated there was no reason not to consider the chatbot system, with this Figure evenly distributed across all age groups (Figures 16 to 18).

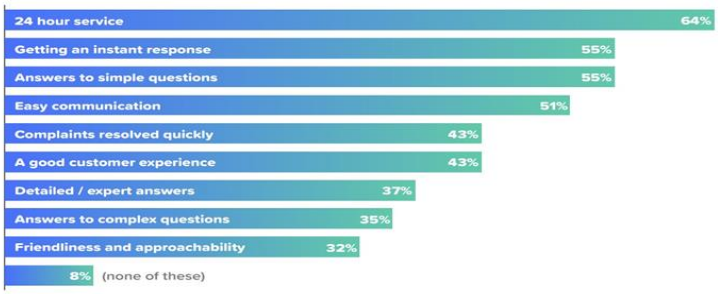

Figure 16: Result of the “Why will you consider a chatbot?”

The results show that people are confused in various ways; the primary reason is that they lack full confidence in the technology. In other words, there is a lack of awareness among people that prevents them from understanding the potential benefits of this technology.

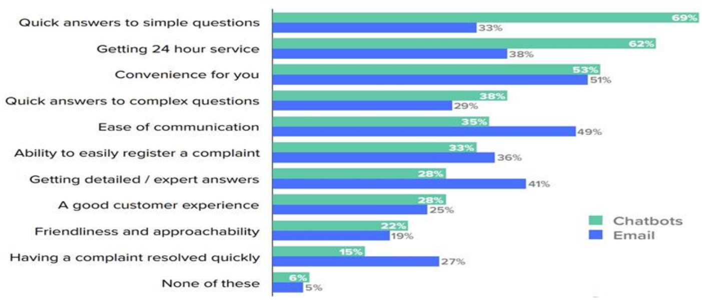

Figure 17: Chatbot vs email

The primary consideration for any business when adopting a chatbot system is to ensure instant replies. For any other complicated situation, a chatbot can refer to a human assistant. We have asked the participant to compare chatbots with email, chatbots with apps, and chatbots with phones. The question was.

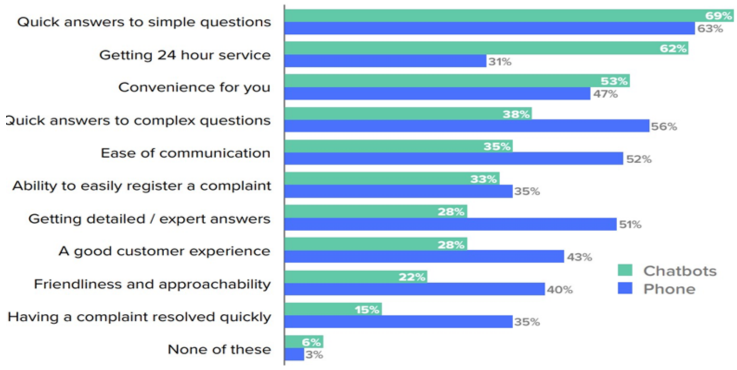

Figure 18: Chatbot vs phone

4.1.6. Chatbot vs App

In business communication, chatbots are still in the process of replacing phone and email use. The results above show that participants preferred the chatbot over the app in five ways, including instant replies, 24-hour access, detailed explanations, and solutions to complex questions. Furthermore, when compared to the email section, the participant preferred the chatbot for instant replies and 24-hour service (Figure 19).

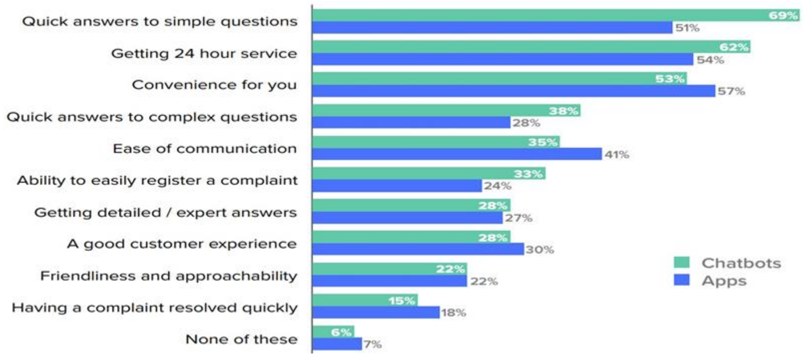

Figure 19: Chatbot vs app

When comparing results with the phone, it is worth noting that participants preferred the phone for an instant solution over the chatbot. Email and phone are the primary channels for resolving complex issues and registering complaints. The survey indicates that chatbots are still in the early stages of development and improvement in processing complex queries; however, they remain one of the best media for business communication, particularly for instant replies and 24-hour service.

5. Artefact Design

5.1. Overview of Design

In this paper, we had deliberation on the system design and its process. We also include diagrams that provide an overview of the chatterbot's design. A Gantt chart is also explained in this section to illustrate the time frame we have used throughout the development process.

5.2. Basic Chatbot Diagram

The chatbot is customer service-focused, so the datasets, such as question-answer pairs and question-to-question, are based on one-to-one conversations. However, some models may be more applicable across domains. The chatbot is designed to be general, applicable across various domains and datasets, including question-answer and question-to-question datasets. Different methods and models have been explained, and based on the discussion in prior sections, the chatbot has been developed using machine learning, which will be discussed in detail in later sections of this report (Figure 20).

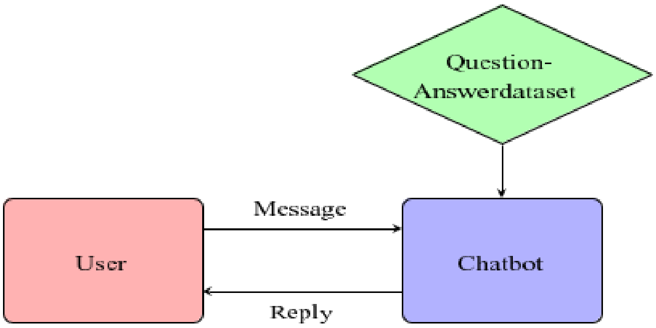

Figure 20: Basic diagram of a chatbot

This diagram provides a comprehensive overview of the machine learning chatbot model's functionalities.

5.3. Flow Diagram Design

A Flow Diagram is a visual representation of the sequence of steps and decisions needed to perform a process. Each step in the sequence is noted with the diagram shapes. Steps are linked by connecting lines, so that anyone viewing the Diagram can logically follow the process from beginning to end. It is a powerful tool for effectively designing and constructing the necessary steps in the process (Figure 21).

Figure 21: Flow diagram of your chatbot

Notice the diagram with different shapes [17]. Here, we can see that the user connects directly to the chat platform, either via a third-party platform or an AE platform, such as Facebook Messenger, Slack, or another service. That platform connects with the machine learning algorithm. Machine learning algorithms and code connect to the API and database, and they act accordingly.

5.4. Activity Diagrams

The activity diagram depicts the flow of information. Additionally, it provides an overview of all the processes that contribute to the chatbot. The diagrams provide a visual representation of users' behaviour when they try to communicate with the chatbot. The diagrams also demonstrate the chatbot's distinctive workability. For example, after a machine learning algorithm evaluates the user’s input, the server steps in and handles the POST endpoints. Also, the server approves the requests that the machine learning algorithm has already processed. Afterwards, the server should determine the users' intent (Figure 22). When the user starts a conversation with the Chatbot, if the question relates to a conversation already in the machine learning algorithm, the Chatbot will return a response from the server. If not, it will suggest a response or request additional information.

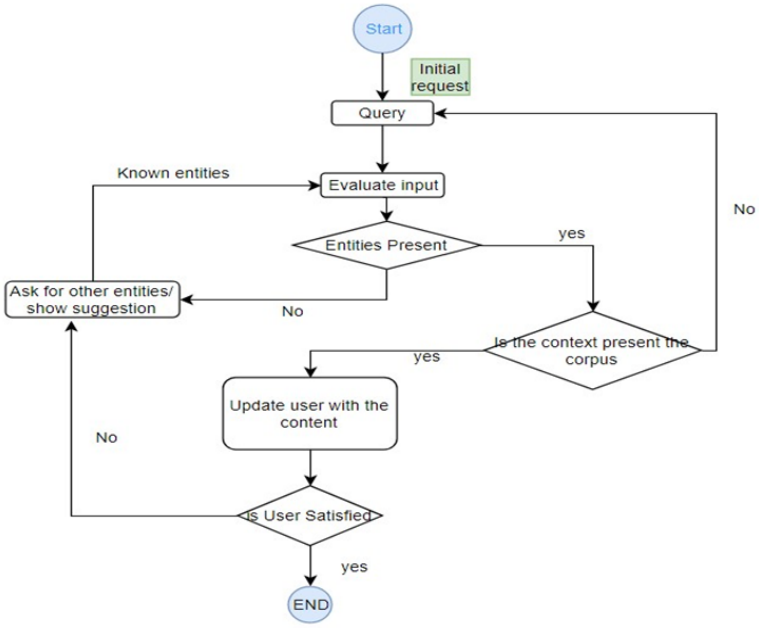

Figure 22: Activity diagram

5.5. Sequence Diagrams

The diagram is known as a sequence diagram. It represents the procedure a user follows when they ask the chatbot for information. This is a fundamental process involved at the beginning. In the form of a request, the server initiates the arrival from the chatbot machine learning algorithm engine. Later, the server is involved in determining the actions needed to analyse the questions it has already received. Afterwards, the server will take the necessary steps if the action matches the customer service. After the API returns data, data encoding begins, and the resulting response is delivered back to the machine learning algorithm. The user can then view the chatbot (Figure 23).

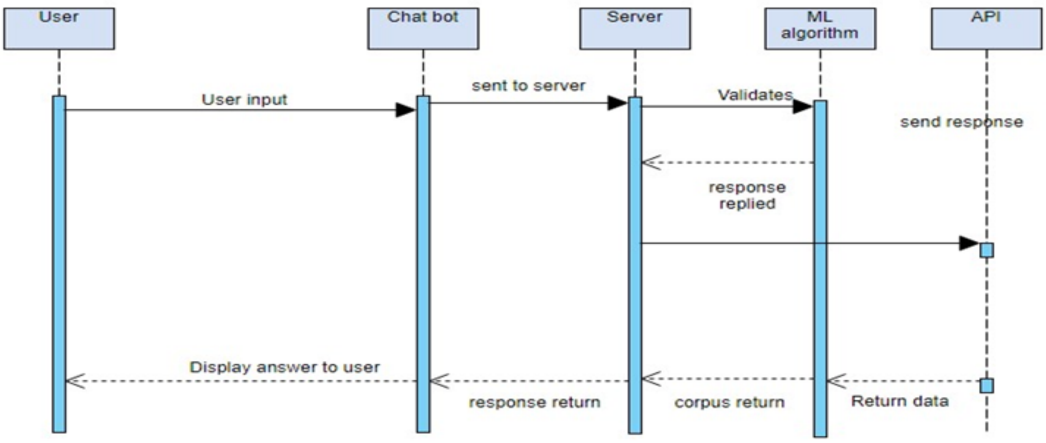

Figure 23: Sequence diagram

5.6. Process Flow Diagram

An untrained instance, lacking prior knowledge of how to engage in conversation, learns how to converse from the training module of the chatterbot library (Figure 247.

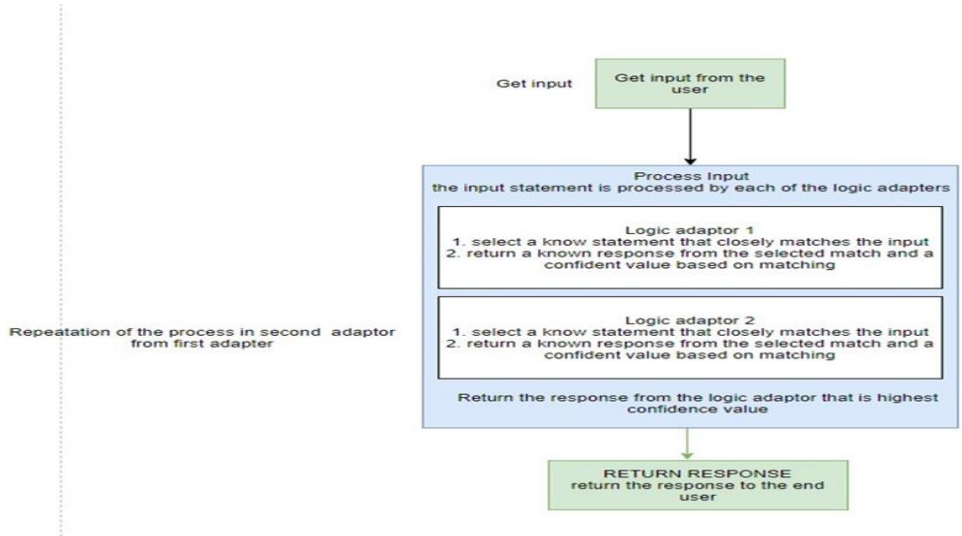

Figure 24: Process flow diagram

5.7. Architectural Design

The figure portrays an architectural design of the proposed chatbot (Figure 25).

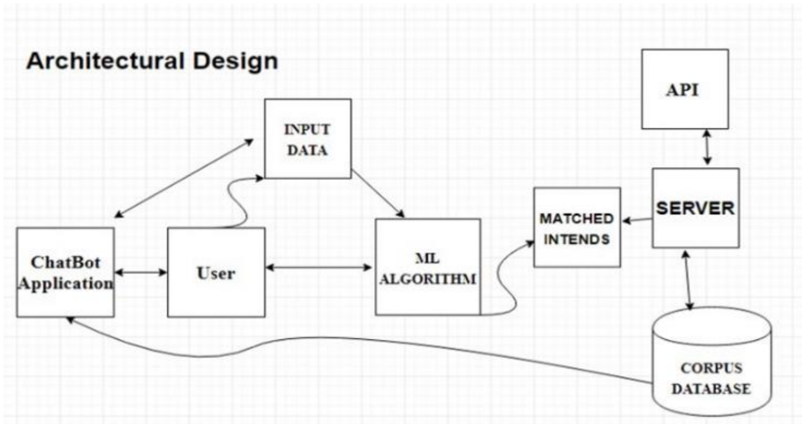

Figure 25: Architectural design

Through the web client, users will communicate with the chatbot. The communication medium between the user and the chatbot will be natural language. CSS, HTML5, and JavaScript have been used to implement the client-side of this application. Besides, this diagram shows us how to utilise in executing the web client, chatbot. The server is entitled to receive the requested data through HTTPS POST from a machine learning algorithm. The route can be easily stated as the endpoint, as a machine learning algorithm can post a real-time payload. In addition to that responsibility, the server also receives data when a user has already established intent. Therefore, the server can receive data posted by different services at any time, as noted in Baeza-Yates and Frakes [4]. This diagram illustrates the chatbot's familiarisation with intents and entities through various training sessions, and is dedicated to utilising a machine learning algorithm. The chatbot will use the intent map to route users' utterances to the consolidation of related words. The Figure illustrates the information flow, which corresponds to a user's intent. It is responsible for checking whether the intent was posted, since it has been trained to do so. Later, the data should be posted on the server. A response is then originated, which is custom in nature.

5.8. Project Management

Project management is essential for any project that is conducted in a supervised environment. We have used a Gantt chart to illustrate the timeline utilised in the development of this paper.

5.8.1. Gantt Chart

A Gantt chart has been created to provide a project timeline. According to the following Gantt chart given below, the paper is expected to be completed on time. Similarly, from the chart, the breakdown of different tasks, the order in which they should be completed before and after, and the allocation of each task can be known (Figure 26).

Tasks	Duration	Start Date	End Date
Job InterView	1	25/03/2019	25/03/2019
Job Duration	119	01/04/2019	22/08/2019
Project Proposal	1	24/03/2019	24/03/2019
Placement Acceptance by university	1	02/04/2019	02/04/2019
Discuss the Job role with Work Supervisor	1	03/04/2019	03/04/2019
Design of poster,leaflet / writing content etc	112	10/04/2019	22/08/2019
Define and outline the problem	21	17/04/2019	08/05/2019
survey	28	13/05/2019	05/06/2019
Interview	35	10/06/2019	19/06/2019
Requirement Analysis	42	24/06/2019	01/07/2019
Litreture Review	49	01/07/2019	29/07/2019
Feasibility Study	56	29/07/2019	05/08/2019
Anaysis, requirements and specifications	63	05/08/2019	02/09/2019
chatbot design and development	70	03/07/2019	23/10/2019
Implementation	77	19/08/2019	08/10/2019
Testing	84	09/10/2019	23/10/2019
Evalutation	91	21/10/2019	28/10/2019
Finalizing of Whole project	98	28/10/2019	04/11/2019
Report Writing	105	22/08/2019	15/11/2019

Figure 26: Gantt chart

The Gantt chart illustrated the time spent during the entire project period, from the beginning of the placement to the end of the report writing. It also clearly shows the duration taken for a particular task and the start date to the end date. It also shows which task took the most time. We can see that chatbot design and development, along with implementation, took longer than any other tasks (Figure 27).

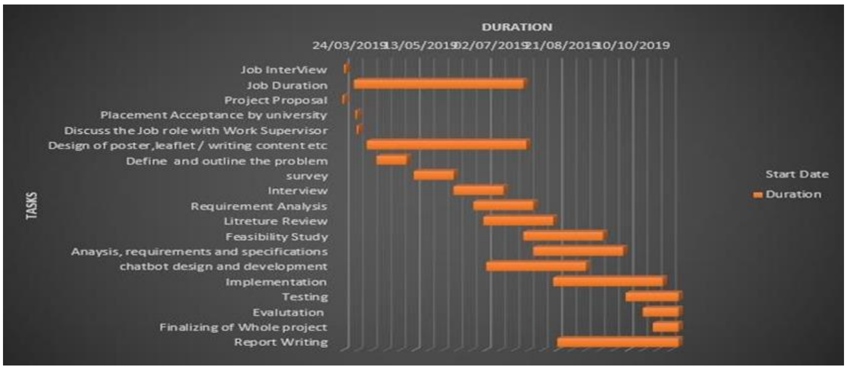

Figure 27: Graph Gantt chart

5.8.2. Risk Management

To mitigate the risks and prevent the consequences of any project, it is essential to identify and manage the associated risks. It is a crucial step for the overall success of any paper. Analysing and understanding the risks and repercussions at the very beginning of the paper minimises their occurrence. It gives us time to take action accordingly in the event of an actual risk occurrence. For our paper, we frequently back up the code to prevent the loss of hard work and algorithms. Using Google Drive made the report and important documents accessible for backup. We are therefore prepared for an unplanned day.

6. Implementation and Testing

6.1. Overview of Implementation and Testing

The software environment and context should be considered when selecting a programming language. In this case, Python has been chosen as the primary programming language for the chatbot. It is a choice between the future software developer (who will work after me) and the execution of the learning catalogue for the given machine. Whilst searching for a chatbot for the implementation phase, we tested a couple of others, but none worked well in our environment. [Appendix F, Appendix G, and Appendix H].

6.2. Installation

For the chatterbot to work, we need to install some prerequisite software on the system. We need to install PyPI, pip, Chatterbot, and some other dependencies (Figures 28 to 30).

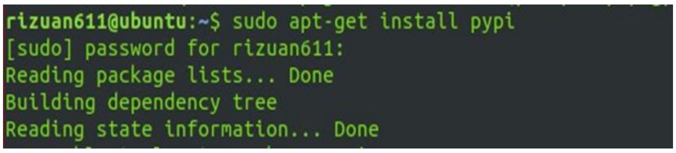

Figure 28: PyPI installation

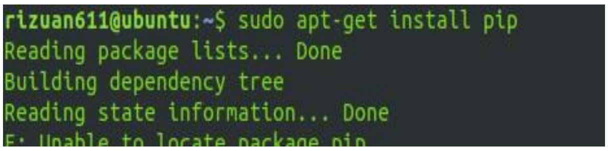

Figure 29: Pip installation

```
Sudo pip3 install chatterbot.
```

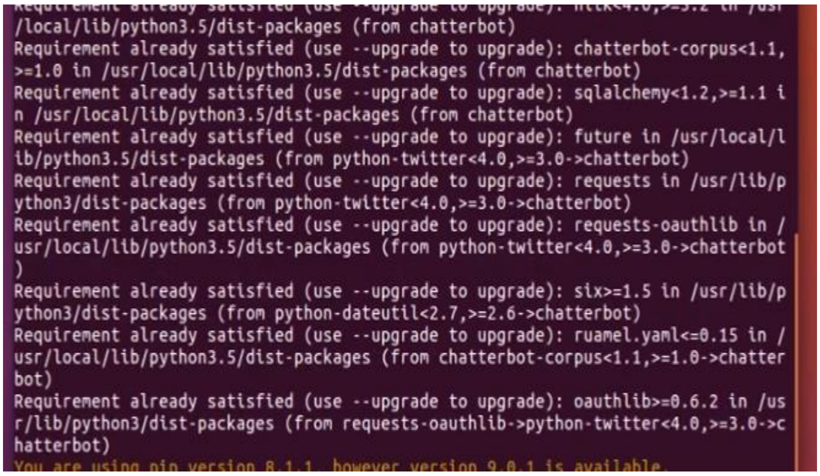

Figure 30: Chatterbot installation

We have installed the chatbot on our computer.

6.3. Libraries and Modules

In terms of libraries we used to build a chatbot, we considered some that helped us create an intelligent chatbot. The first and essential library we used was the Chatterbot library, which helps build an autonomous chat system for any purpose. Its response is based on the user's input. It utilises machine learning algorithms to generate various types of responses. Another important use is Pandas, which allows developers to organise and manage data efficiently. This library is used to manipulate and retrieve data from large amounts of information. Finally, a simple but important module used in the paper was the random module. It generates a random number of outputs from a range specified in the same argument.

6.4. Preparing for the Dependencies for the Chatbot

After creating a Python virtual environment, we need to install the chatterbot library. The following commands help us do so.

```
Pip install chatterbot
Pip install chatterbot_corpus
```

It is better to upgrade whenever we install something. So, the command is:

```
pip install--upgradechatterbot_corpus
pip install--upgradechatterbot
```

6.5. Importing Classes

For the chatterbot, we will need to import two classes for now, so the command is as follows.

```
From chatterbot import ChatBot
From chatterbot.trainers import ListTrainer
```

6.6. Training the Chatbot

The ChatterBot includes various tools to simplify the training process. The dialogue process involves adding different examples to the chatbot database. Known responses are presented, which are created based on the data structure. The entries are created when the trainer provides a dataset. Afterwards, the responses can be represented appropriately (Figure 31).

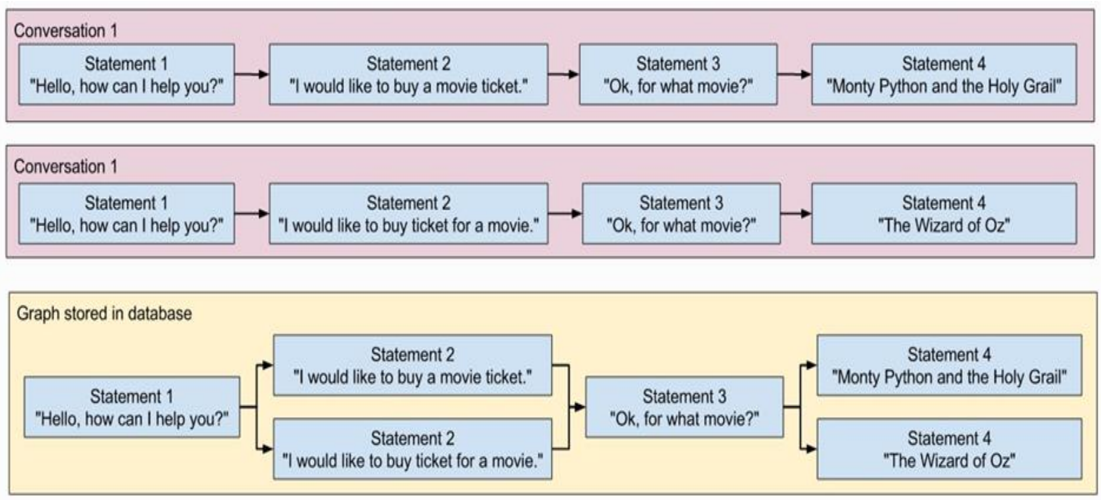

Figure 31: Training phase in the graph

Several training classes are already built into the ChatterBot. The utilities can vary, such as updating the chatbot's database knowledge or granting permission to train the bot. It should be done based on corpus training data that should be pre-loaded by nature. In addition, one can devise their intended training class. This can be recommended if one intends to train the bots using data stored in the system. The list is given as follows.

6.6.1. Training Classes

A built-in training class accompanies Chatterbot. We can create our training system in the chatbot to train the instance. Additionally, it is possible to devise a new one as needed. It is vital to call the train, which has been started with the chatbot, to train it. At the beginning of the development phase, the instance is not intelligent. It has no clue about the human conversation. The chatterbot has some tools that can process the training phase. We can train the chatterbot in two main ways.

- Training on the list data.
- Training with corpus data.

To learn how the chatbot works, we tested it on both systems. For implementation, we utilised the corpus data system to train it. When the trainer provides training to the chatbot, it achieves the necessary knowledge to respond with the statement inputs and responses as represented in the training phase.

6.6.2. Training Via List Data

```
chatterbot.trainers.ListTrainer(chatbot, **kwargs)[source]
```

It grants the chatbot permission to train on the listed types of strings that represent a conversation. Regarding the training process, it is essential to provide a comprehensive list of statements. The organisation of the statements is done based on a particular conversation. For example, if one intends to run a bot by saying, “Hello”, the chatterbot’s response to the statement will result in “Hi there!” or “Greetings!” (Figure 32).

chatbot.py

```
chatbot = ChatBot('Training Example')
```

train.py

```
from chatbot import chatbot
from chatterbot.trainers import ListTrainer

trainer = ListTrainer(chatbot)

trainer.train([
    "Hi there!",
    "Hello",
])

trainer.train([
    "Greetings!",
    "Hello",
])
```

Figure 32: Training via list data

A comprehensive list of training conversations can be provided, with each item identified as a response to all preceding items (Figure 33).

```
trainer.train([
    "How are you?",
    "I am good.",
    "That is good to hear.",
    "Thank you",
    "You are welcome.",
])
```

Figure 33: Training via long list data

6.6.3. Training with Corpus Data

chatterbot.trainers.ChatterBot	CorpusTrainer(chatbot, **kwargs)[source]

This can allow the chatbot to be trained using different data from the dialogue corpus, ChatterBot. ChatterBot interacts with the corpus data through the utility module, which helps ensure smooth training for the bot. To do this, it is imperative to specify the intended corpus module (Figure 34).

chatbot.py

```
chatbot = ChatBot('Training Example')
```

train.py

```
from chatbot import chatbot
from chatterbot.trainers import ChatterBotCorpusTrainer

trainer = ChatterBotCorpusTrainer(chatbot)

trainer.train(
    "chatterbot.corpus.english"
)
```

Figure 34: Training via corpus

We have trained the chatterbot to be just a simple chatbot. We feed the data into the system in a YAML file. A sample of the greeting feeding is as follows (Figure 35).

```yaml
categories:
- conversations
conversations:
- - Good morning, how are you?
  - I am doing well, how about you?
  - I'm also good.
  - That's good to hear.
  - Yes it is.
- - Hello
  - Hi
  - How are you doing?
  - I am doing well.
  - That is good to hear
  - Yes it is.
  - Can I help you with anything?
  - Yes, I have a question.
  - What is your question?
  - Could I borrow a cup of sugar?
  - I'm sorry, but I don't have any.
  - Thank you anyway
  - No problem
- - How are you doing?
  - I am doing well, how about you?
  - I am also good.
  - That's good.
- - Have you heard the news?
  - What good news?
- - What is your favorite book?
  - I can't read.
  - So what's your favorite color?
  - Blue
- - Who are you?
  - Who? Who is but a form following the function of what
  - What are you then?
  - A man in a mask.
  - I can see that.
```

Figure 35: Corpus data

If we test the chatbot, we can see it responds appropriately (Figure 36).

```
List Trainer: [####################] 100%
You: hi
Bot:  hello

You: how are you
Bot:  I'm fine thanks for asking

You: awesome
Bot:  Yes, how's yours?

You: your what?
Bot:  It was great.

You: What was great
Bot:  Awesome

You: Are you crazy?
Bot:  I'm fine thanks for asking

You: Did you understand this?
Bot:  No, which one?

You:
```

Figure 36: After corpus training

We have built our own corpus to train the chatbot, making the conversation more human-like (Figure 37).

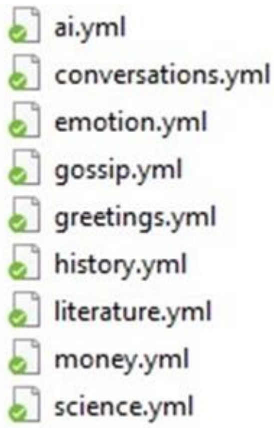

Figure 37: Different corpus

These are the other files that we have used in the chatterbot.

6.7. Integrating the Chatbot in the Website

After building up the chatbot, we need to deploy it to the E20Minicabs website. There are various web-based frameworks for generating Python projects. The best examples of these projects are Django and Flask. The chatterbot's design follows quite user-friendly steps in terms of its compatibility with the allocated platform. The initial requirement for deploying or running a chatterbot in a web-based application is the ability to receive and send data. It can be done in several ways, such as HTTPS and web sockets. To integrate a chatterbot into our web application, we utilised Django. We need to install Django and ChatterBot so the Python program can run on a web server.

```
pip install Django chatterbot
```

We also need to migrate the database, so the command is:

```
python manage.py migrate django_chatterbot
```

6.8. Web Interface

In the following picture, we can see that the web integration is done. The chatterbot is in green colour at the bottom right of the page (Figure 38).

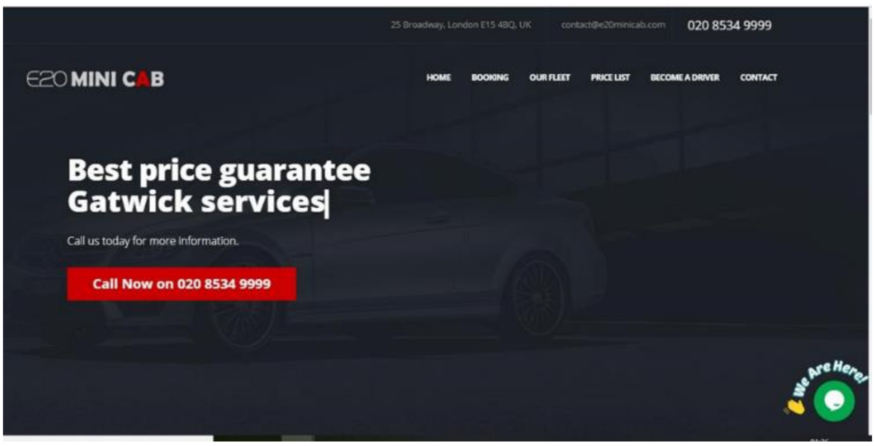

Figure 38: Web interface

We have designed the chatbot to pop up when the user spends more than 15 seconds on the website (Figure 39).

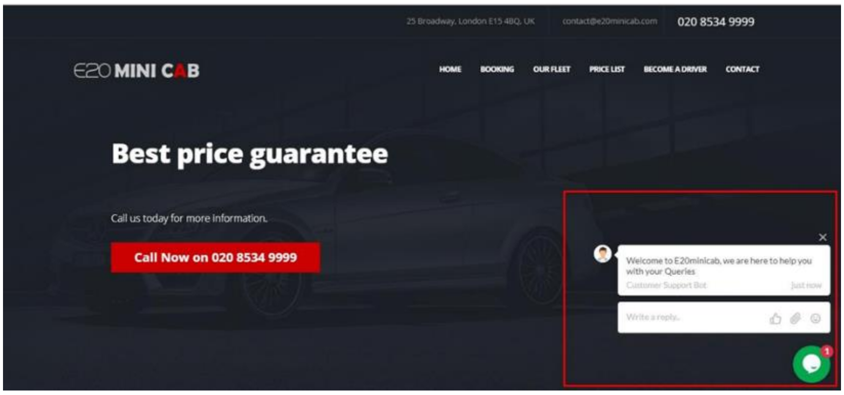

Figure 39: Chatbot appears

The following screenshot shows the chatbot popping up automatically when the web application is launched, and it asks the user for any assistance required, then continues chatting (Figure 40).

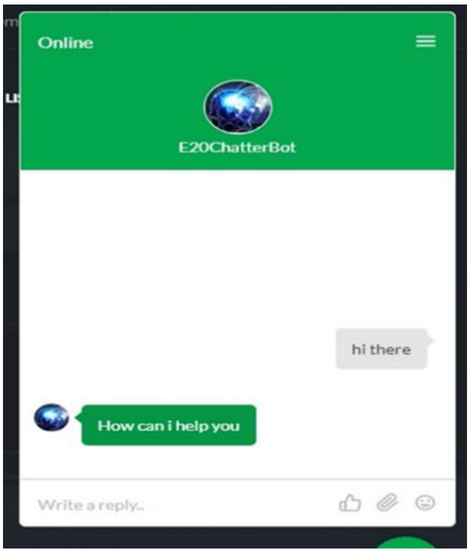

Figure 40: Conversation with chatbot

6.9. Testing

The testing phase of SDLC is a significant part of the system development lifecycle, as it demonstrates the performance of the implemented system. It also outlines the system's limitations and the extent of its success. At this stage of the SDLC, various bugs and errors affecting the system's overall performance are identified and recorded for further improvement. This feature ensures the efficient performance of the chatbot system. We have performed a variety of tests in multiple ways to check the effectiveness of the implemented system. While considering the testing phase of our SDLC for the chatbot system, the following are some of the tests we have done to check the performance of our system:

- White box testing
- Black box testing
- Stress testing

6.9.1. White Box Testing

White box testing is an effective method for testing any implemented system based on its internal structure and programming. It is also called glass-box testing, in other words. As the implementation phase is complete, we are now considering software for testing. It will ensure the system's operational performance. There are several ways to test the application code to determine whether it is working as expected. We have used a couple of software tools to analyse code, and at certain points we have discovered errors; however, we have debugged them. We have coded the chatterbot software using PyCharm. When we run the program, we encounter a couple of errors, as shown in the screenshot below (Figure 41).

Figure 41: Errors

After debugging, our PyCharm showed the results without error. We have done troubleshooting on various online media, and ultimately, the software started working fine. The following screenshot shows the results of white-box testing after debugging (Figure 42).

Figure 42: After debugging

6.9.2. Black Box Testing

Black-box testing is primarily concerned with the system's behaviour. Because this testing system is followed by input and output values, the information available about the system is limited. If there are issues or errors in the algorithm, this test will display them. To verify the system's functionality, we conduct tests on an individual task basis, using each test to evaluate a single function. Various high-level function tests differ from domain testing. Functional testing of our paper demonstrates the intended operation of our chatbot system. As shown in the picture below, when the user launches the website, the chatbot automatically appears, asking for help, and the conversation can then continue. This test proves the normal function of the implemented chatbot system (Figure 43).

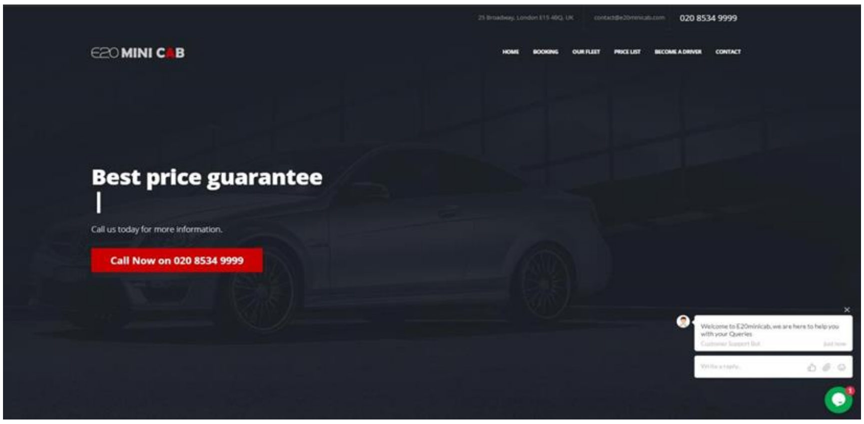

Figure 43: Chatbot with the web interface

After this, we conducted additional functional testing to assess the chatbot system's conversational behaviour. The screenshot below displays the system's response to the user input, allowing the conversation to continue. This test demonstrates the system's functional behaviour in responding to the user (Figure 44).

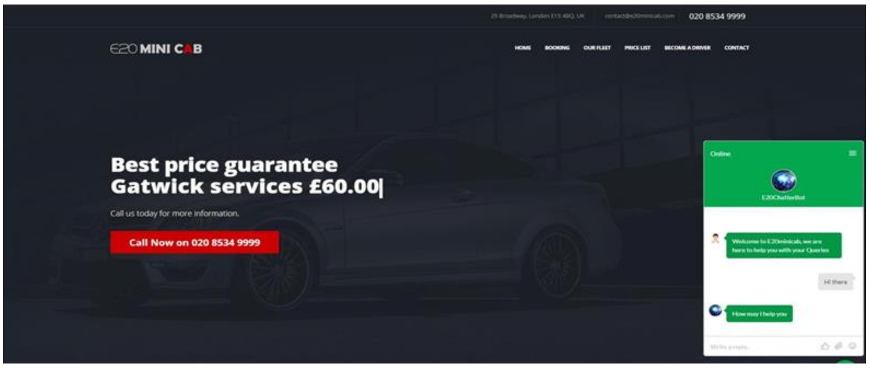

Figure 44: Functionality of chatbot

The screenshot below illustrates that the chatbot fails to comprehend questions or queries unrelated to the company and has not learned from them (Figure 45).

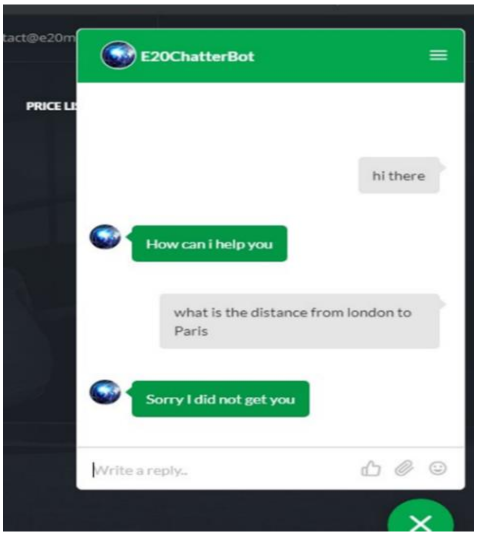

Figure 45: Response to the unknown query

6.9.3. Stress Testing

This type of testing involves pushing the program beyond its limits to monitor system behaviour under extreme conditions (stress).

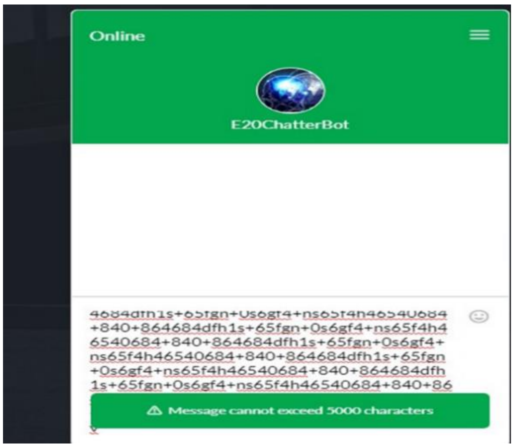

Figure 46: Stress testing

There is a chance the system will not operate as usual, which can help observe its behaviour in the event of any issues or errors. Understanding the behaviour trend can help assess the system's vulnerability in extreme cases, and thus various test techniques can be applied to reduce the risk of potential issues. It required a right-hand approach to troubleshooting the system to resolve the error (Figure 46). In our paper, we performed stress testing by entering over 5000 characters to test the system's ability to handle a specific number. After testing, the picture above shows the message once it reaches 5000 characters. This test result indicates that this system cannot support characters with a code point above 5000 and also disables the “Enter” key.

6.10. Error Handling

During software development, we often encounter various errors. The following are some of the types of errors:

- The language's syntax caused the error.
- The semantic error can be caused by inefficient use of a program statement.
- A logical error can occur if the program statement is not satisfied. Although the program runs without error, the expected outcome is not generated.

Based on the errors mentioned above, we have distinguished:

- **Compile-time error:** Semantic and Syntax errors are distinguished by the compiler.
- **Runtime error:** It includes semantic (dynamic) and logical errors that are not distinguished by the compiler (Figure 47).

Figure 47: Error handling

After integrating the chatbot into the website, we found that it was not responding to questions quickly enough. We conducted troubleshooting and found that the “response_time” parameter was high, resulting in a long response delay. There was also an incident where that chatterbot was not answering as expected, and we were giving a random response from the data list. We have found that the 'chatterbot. Logic.BestMatch was misconfigured, and then we corrected the system.

7. Conclusion and Future Work

7.1. Conclusion Overview

The primary objective of this paper for the company was to develop a chatbot capable of generating responses to complex customer questions. The aim was to give well-defined, logical replies. We have discussed tools and techniques for implementing the chatbot system and selecting the most appropriate ones to fulfil the company's requirements. The primary concerns were the NLP, vector space model, and machine learning model. The following are the deliverables of this paper:

- The successful operation of the chatbot system.
- Effective use of machine-learning techniques in the chatbot system.
- Evaluation of the chosen technology.

The following criteria have been successfully met, yet the chatbot's answering quality is not very high. The following are some reasons for the low-quality answers provided by the chatbot system implemented on the company website. There is a gap in the machine learning model for the Q2Q datasets used in the chatbot implementation. This is due to the smaller datasets of the questions and answers, which do not provide all the answers to a variety of questions. The result can be unexpected when a user inputs a question that the dataset does not know. On the other hand, if a well-justified and well-defined answer from the chatbot follows the question that matches our dataset. It is quite logical that if more than half of the user-input questions do not align with our chatbot datasets, the results will be the worst. Another justification for less-defined answers is the specific datasets for individual questions. Because the user-input question is defined only for the inevitable question, it sometimes conflicts with the new user input.

7.2. Limitations

Some limitations were discovered during testing, stemming from the inability to retrieve the correct answers to the user's input question. We have identified logs indicating this issue. A user expects higher efficiency in getting satisfying answers to their questions, which may span multiple topics and areas; however, this variety of information may not be available in the database. The implemented chatbot lacks memory, so if the user asks similar questions multiple times, the answer remains the same. We have not used Anaphora at this stage, which could be useful to the user, and the system can refer to the previous question. It requires memory for its operation. Another limitation is its inability to identify spelling errors, so the user has to manually correct the spelling and re-enter the question. The system does not work due to errors in noun and verb usage in the sentences, so users must follow the correct pattern. If it is submitted in the wrong order, the user will have to resubmit the question with the correct noun-verb order in the sentence. For our paper, the chatbot system is indeed useful for the company; however, it still needs improvement in its user interface to be more user-friendly and efficient.

7.3. Future Work

Based on our research and the implementation of the chatbot system for the company, we have discovered the following future work that can cause a significant improvement in this system:

- Firstly, our paper covers limited question-answer datasets that comprise smaller domains. We have considered around 150 questions and answers and realised that this number is insufficient for the customer on a larger scale. This can be improved by extending datasets in the smaller domains.
- This chatbot is limited to following only the specific answer, so the issue arises when a new question is entered in a similar category. For future improvement, it is worth rewriting answers so they can be used for multiple questions.
- This chatbot system is more effective in answering questions than in initiating conversations with users by providing relevant information. This can be improved in the future by integrating more functionality into the chatbot system, enabling it to engage in conversations rather than simply replying to specific questions.
- For the chatbot system that is based on machine learning techniques, it could be beneficial to add hand-coded rules to answer matching questions, i.e., “what is the distance of Heathrow from London?”. This can be quite useful, as it means fewer complicated questions to answer.
- The general operation of the chatbot system for machine learning typically takes 15 seconds; however, a similar chatbot used by Scala processes an enquiry in a millisecond. For future improvement, it would be beneficial to run the chatbot in a Python web application using machine learning techniques.
- As mentioned in the literature review, the advantages of considering a combined chatbot system that uses both NLP and a vector-space model are discussed. This requires a significant amount of training data; however, in the future, it is worth considering implementing this chatbot system by combining these models.

The research on the Machine Learning approach has been very time-consuming. Our success in this paper demonstrates that a chatbot can effectively handle complex question-and-answer retrieval.

Acknowledgement: The author gratefully acknowledges the support and resources provided by Northumbria University, which greatly contributed to the successful completion of this research work.

Data Availability Statement: The datasets generated or analyzed during this study are available from the corresponding author upon reasonable request.

Funding Statement: This research was conducted independently, without any external financial assistance or institutional funding.

Conflicts of Interest Statement: The author declares that there are no conflicts of interest associated with this research. All referenced materials and sources have been properly cited.

Ethics and Consent Statement: This study adheres to the highest ethical research standards, and informed consent was obtained from all participants involved in the study.

References

1. A. V. Aho and J. D. Ullman, “The Theory of Parsing, Translation, and Compiling,” *Prentice-Hall*, Upper Saddle River, New Jersey, United States of America, 1972.

2. T. Akiba, K. Itou, and A. Fujii, “Question answering using ‘common sense’ and utility maximization principle,” *in Proc. NTCIR-4*, Tokyo, Japan, 2004.
3. A. D. Angelica, “How Watson works: A conversation with Eric Brown, IBM Research Manager,” Kurzweil Accelerating Intelligence, 2011. Available: https://www.thekurzweillibrary.com/how-watson-works-a-conversation-with-eric-brown-ibm- researchmanager#:~:text=Rather%2C%20we%20use%20natural%20language,way%20of%20expressing%20those%20questions [Accessed by 20/08/2024].
4. R. Baeza-Yates and W. B. Frakes, “Information Retrieval: Data Structures & Algorithms,” *Prentice-Hall*, Upper Saddle River, New Jersey, United States of America, 1992.
5. S. Banerjee and A. Lavie, “METEOR: An automatic metric for MT evaluation with improved correlation with human judgments,” *in Proc. ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, Ann Arbor, Michigan, United States of America, 2005.
6. A. Barr, “Google plans new, smarter messaging app,” The Wall Street Journal, 2015. Available: https://www.wsj.com/articles/google-plans-new-smarter-messaging-app-1450816899 [Accessed by 07/08/2024].
7. Y. Bengio, R. Ducharme, P. Vincent, and C. Janvin, “A neural probabilistic language model,” *J. Mach. Learn. Res.*, vol. 3, no. 2, pp. 1137–1155, 2003.
8. A. Berger, R. Caruana, D. Cohn, D. Freitag, and V. Mittal, “Bridging the lexical chasm: Statistical approaches to answer finding,” *in Proc. 23rd Annu. Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, Athens, Greece, 2000.
9. G. Corrado, “Computer, respond to this email,” *Google Research Blog*, 2015. Available: https://research.google/blog /computer-respond-to-this-email/ [Accessed by 01/08/2024].
10. D. Cutting, J. Kupiec, J. Pedersen, and P. Sibun, “A practical part-of-speech tagger,” *in Proc. 3rd Conf. Appl. Nat. Lang. Process.*, Trento, Italy, 1992.
11. A. Descoins, “Chatbots and automated online assistants,” *tryolabs* 2015. Available: https://tryolabs.com/blog/2015/0 7/23/cha tbots-and-automated-online-assistants/ [Accessed by 08/08/2024].
12. K. M. Fauske, “Example: Neural network,” *texample*, 2006. Available: http://www.texample.net/tikz/exampl es/neural-network/ [Accessed by 24/08/2024].
13. D. Ferrucci, E. Brown, J. Chu-Carroll, J. Fan, D. Gondek, A. A. Kalyanpur, A. Lally, J. W. Murdock, E. Nyberg, J. Prager, N. Schlaefer, and C. Welty, “Building Watson: An overview of the DeepQA project,” *AI Magazine*, vol. 31, no. 3, pp. 59–79, 2010.
14. T. Hastie, R. Tibshirani, and J. Friedman, “The Elements of Statistical Learning,” *Springer Series in Statistics. Springer*, Berlin, Germany, 2001.
15. S. Ghosh, O. Vinyals, B. Strope, S. Roy, T. Dean, and L. Heck, “Contextual LSTM (CLSTM) models for large scale NLP tasks,” *arXiv preprint arXiv:1602.06291*, 2016. Available: https://arxiv.org/abs/1602.06291 [Accessed by 01/08/2024].
16. B. Goertzel and C. Pennachin, “Artificial General Intelligence,” *Springer*, Berlin, Germany, 2007.
17. D. C. Gondek, A. Lally, A. Kalyanpur, J. W. Murdock, P. A. Duboue, and L. Zhang, “A framework for merging and ranking of answers in DeepQA,” *IBM J. Res. Develop.*, vol. 56, no. 3-4, pp. 1–12, 2012.
18. J. Gu, Z. Lu, H. Li, and V. O. K. Li, “Incorporating copying mechanism in sequence-to-sequence learning,” *arXiv preprint arXiv:1603.06393*, 2016. Available: https://arxiv.org/abs/1603.06393 [Accessed by 25/08/2024].
19. S. M. Harabagiu, D. I. Moldovan, M. Pasca, R. Mihalcea, M. Surdeanu, R. C. Bunescu, R. Girju, V. Rus, and P. Morarescu, “Falcon: Boosting knowledge for answer engines,” *egr*, 2000. Available: https://www.egr.msu.edu/~jcha i/QAPapers/falcon.pdf [Accessed by 13/08/2024].
20. S. Jean, K. Cho, R. Memisevic, and Y. Bengio, “On using very large target vocabulary for neural machine translation,” *arXiv preprint, arXiv:1412.2007*, 2014. Available: http://arxiv.org/abs/1412.2007 [Accessed by 22/08/2024].
21. R. Kadlec, M. Schmid, and J. Kleindienst, “Improved deep learning baselines for Ubuntu corpus dialogs,” *arXiv preprint arXiv:1510.03753*, 2015. Available: https://arxiv.org/abs/1510.03753 [Accessed by 05/08/2024].
22. A. Kannan, K. Kurach, S. Ravi, T. Kaufmann, A. Tomkins, B. Miklos, G. Corrado, L. Lukacs, M. Ganea, P. Young, and V. Ramavajjala, “Smart Reply: Automated response suggestion for email,” *arXiv preprint, arXiv:1606.04870*, 2016. Available: https://arxiv.org/abs/1606.04870 [Accessed by 31/08/2024].
23. H. Kim and J. Seo, “Cluster-based FAQ retrieval using latent term weights,” *IEEE Intell. Syst.*, vol. 23, no. 2, pp. 58–65, 2008.
24. T. K. Landauer, “Latent Semantic Analysis,” *Wiley Online Library*, Hoboken, New Jersey, United States of America, 2006.
25. J. Li, M. Galley, C. Brockett, J. Gao, and B. Dolan, “A diversity-promoting objective function for neural conversation models,” *arXiv preprint arXiv:1510.03055*, 2015. Available: https://arxiv.org/abs/1510.03055 [Accessed by 03/08/2024].
26. J. Lin, D. Quan, V. Sinha, K. Bakshi, D. Huynh, B. Katz, and D. R. Karger, “What makes a good answer? The role of context in question answering,” *in Proc. IFIP TC13 Int. Conf. Human-Computer Interaction (INTERACT ’03)*, Zurich, Switzerland, 2003.

27. C.-W. Liu, R. Lowe, I. V. Serban, M. Noseworthy, L. Charlin, and 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. Available: https://arxiv.org/abs/1603.08023 [Accessed by 02/08/2024].
28. R. Lowe, N. Pow, I. Serban, and J. Pineau, “The Ubuntu dialogue corpus: A large dataset for research in unstructured multi-turn dialogue systems,” *in Proceedings of the 16th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, Prague, Czech Republic, 2015.
29. M.-T. Luong and C. D. Manning, “Stanford neural machine translation systems for spoken language domains,” *in Proc. 12th Int. Workshop Spoken Lang. Transl. (IWSLT): Evaluation Campaign*, Da Nang, Vietnam, 2015.
30. M.-T. Luong, I. Sutskever, Q. V. Le, O. Vinyals, and W. Zaremba, “Addressing the rare word problem in neural machine translation,” *arXiv preprint arXiv:1410.8206*, 2014. Available: https://arxiv.org/abs/1410.8206 [Accessed by 13/08/2024].
31. C. D. Manning, P. Raghavan, and H. Schütze, “Introduction to Information Retrieval,” *Cambridge Univ. Press*, Cambridge, United Kingdom, 2008.
32. T. Mikolov, W.-T. Yih, and G. Zweig, “Linguistic regularities in continuous space word representations,” *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Atlanta, Georgia, United States of America, 2013.
33. T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean, “Distributed representations of words and phrases and their compositionality,” *arXiv preprint arXiv:1310.4546*, 2013. Available: https://arxiv.org/abs/1310.4546 [Accessed by 10/08/2024].
34. T. Mikolov, K. Chen, G. Corrado, and J. Dean, “Efficient Estimation of Word Representations in Vector Space,” *arXiv preprint arXiv:1301.3781*, 2013. Available: https://arxiv.org/abs/1301.3781. [Accessed by 15/08/2024].
35. A. Mishra and S. K. Jain, “A survey on question answering systems with classification,” *J. King Saud Univ. - Comput. Inf. Sci.*, vol. 28, no. 3, pp. 345–361, 2016.
36. M. Palmer, D. Gildea, and N. Xue, “Semantic role labeling,” *Synth. Lect. Hum. Lang. Technol.*, vol. 3, no. 1, pp. 1–103, 2010.
37. J. Prager, J. Chu-Carroll, E. W. Brown, and K. Czuba, “Question answering by predictive annotation,” *in Advances in Open Domain Question Answering, T. Strzalkowski and S. M. Harabagiu, Eds., Text, Speech and Language Technology, Springer*, Dordrecht, The Netherlands, 2008.
38. K. Yao, G. Zweig, and B. Peng, “Attention with intention for a neural network conversation model,” *arXiv preprint arXiv:1510.08565*, 2015. Available: https://arxiv.org/abs/1501.08565 [Accessed by 17/08/2024].