Sagarmatha College of Science and Technology

(Under the affiliation of Tribhuvan University)

Sanepa, Lalitpur, Nepal

A Project Report on

“Chatbot”

Submitted to:

Department of Computer Science and Information Technology

Submitted by:

Arpan rai	23914/076
Bhabish karki	23917/076

Semester: seventh

Batch: 2076

Date: 2024, March

Sagarmatha College of Science and Technology

Sanepa, Lalitpur, Nepal

(T.U. affiliate)

SUPERVISOR'S RECOMMENDATION

I hereby recommend that this project prepared under my supervision by **Bhabish Karki** (23917/076) and **Arpan Rai** (23914/076) entitled **"Chatbot"** in partial fulfillment of the requirement for the degree of BSc. In computer science and information technology (BSc. CSIT) be processed for evaluation.

…………………………….

Mr. Sobit Thapa

(Supervisor)

Sagarmatha College of Science and Technology

Sanepa, Lalitpur, Nepal

(T.U. affiliate)

CERTIFICATE OF APPROVAL

This is to certify that this project prepared by **Bhabish Karki** (23917/076) and **Arpan Rai** (23914/076) entitled "**chatbot**" in partial fulfillment of the requirements for the degree of BSc. CSIT has been well studied. In our opinion, it is satisfactory in the scope and quality of the required degree.

…………………………………	…………………………………
Program Coordinator	Supervisor
Mr. Pratik Timalsena	Mr. Sobit Thapa

………………………………	………………………………
Internal Examiner	External Examiner
CSIT Department	Prof. Dr. Subarna Shakya
	IOST, Tribhuvan University

ACKNOWLEDGEMENT

We would like to express our special thanks of gratitude to our Supervisor Mr. Sobit Thapa as well as our Program Coordinator Mr. Pratik Timalsena who gave us the opportunity to do this wonderful project on the topic Chatbot, which also helped us in doing a lot of Research and came to know about so many new things. We are really thankful to them.

The members were very dedicated toward the project and surrender their vital time to accomplish the project within the schedule time by sharing knowledge about different aspect to make the project efficient and effective to use.

We would like to acknowledge our gratitude towards our seniors and our honorable teachers who had accelerated and encouraged us during the development of the project by giving conceptual view for which we had no experience. Also like to thank our Head of Department and other faculty members of computer in vigorous support and kind guidance.

We want to thanks all our friends and people who directly or indirectly assisted us in our work until the completion of our Project.

Bhabish Karki (23917/076)
Arpan Rai (23914/076)

ABSTRACT

Chatbots, or conversational interfaces as they are also known, present a new way for individuals to interact with computer systems. Traditionally, to get a question answered by a software program involved using a search engine, or filling out a form. A chatbot allows a user to simply ask questions in the same manner that they would address a human. The most well-known chatbots currently are voice chatbots: Alexa and Siri. However, chatbots are currently being adopted at a high rate on computer chat platforms.

The technology at the core of the rise of the chatbot is natural language processing (“NLP”). Recent advances in machine learning have greatly improved the accuracy and effectiveness of natural language processing, making chatbots a viable option for many organizations. This improvement in NLP is firing a great deal of additional research which should lead to continued improvement in the effectiveness of chatbots in the years to come.

Keywords—NLP, feed-forward, chatbot, RNN

TABLES OF CONTENT

Page No.

LIST OF ABBREVIATIONS

- AI – Artificial Intelligence
- FNN- Feedforward Neural Network
- ML- Machine Learning
- NER- Named Entity Recognition
- NLG- Natural Language Generation
- NLP- Natural Language Programming
- NLU- Natural Language Understanding
- NTLK- Natural Language Tool Kit
- RNN- Recurrent Neural Network
- UML – Unified Modeling Language

LIST OF FIGURES

List of Tables

Chapter 1: Introduction

1.1. Introduction:

At the most basic level, a chatbot is a computer program that simulates and processes human conversation (either written or spoken), allowing humans to interact with digital devices as if they were communicating with a real person. You've probably interacted with chatbot whether you know it or not. For example, you're at your computer researching a product, and a window pops up on your screen asking if you need help. Or perhaps you're on your way to a concert and you use your smartphone to request a ride via chat. The working of chatbots is driven by AI, automated rules, natural language processing (NLP), and machine learning (ML), chatbots process data to deliver responses to requests of all kinds.

There are two main types of chatbots are Task-oriented (declarative) chatbots and Data-driven and predictive (conversational) chatbots. The former type of chatbots includes single-purpose programs that focus on performing one function. Using rules, NLP, and very little ML, they generate automated but conversational responses to user inquiries. Interactions with these chatbots are highly specific and structured and are most applicable to support and service functions—think robust interactive FAQs. The latter type of chatbots are referred to as virtual assistants or digital assistants, and they are much more sophisticated, interactive, and personalized than task-oriented chatbots. These chatbots are contextually aware and leverage natural-language understanding (NLU), NLP, and ML to learn as they go. They apply predictive intelligence and analytics to enable personalization based on user profiles and past user behavior. Apple's Siri and Amazon's Alexa are examples of consumer-oriented, data-driven, predictive chatbots. Chatbots boost operational efficiency and bring cost savings to businesses while offering convenience and added services to internal employees and external customers. They allow companies to easily resolve many types of customer queries and issues while reducing the need for human interaction.

Soon, when AI is combined with the development of 5G technology, businesses, employees, and consumers are likely to enjoy enhanced chatbot features such as faster recommendations and predictions, and easy access to high-definition video conferencing from within a conversation. Our chatbot is designed in a way such that we have defined certain frequently asked questions (FAQs) and trained our chatbot with this data using neural networks, NLP and PyTorch. Proceeding further in the paper we will understand how the chatbot is designed and executed.

1.2. Problem Statement:

In various contexts such as document management, data categorization, labeling tasks are essential for organizing and extracting meaningful information. However, manually labeling data can be time-consuming, error-prone, and resource-intensive. To streamline this process and improve efficiency, there is a need for a basic chatbot that can assist users in labeling tasks by interpreting instructions, generating labels, and facilitating data annotation seamlessly.

The development of a basic label chatbot addresses these challenges by providing an intuitive interface for users to input labeling instructions, receive automated label suggestions, and verify or modify labels as needed. By leveraging natural language understanding and machine learning techniques, the chatbot aims to accelerate the labeling process, improve label consistency, and enhance overall data quality for various applications.

1.3. Objectives:

The basic objectives focus on providing fundamental functionality and delivering a positive user experience while laying the groundwork for potential growth and improvement. Here are some objectives of chatbots.

- Offer straightforward responses to user queries to fulfill their informational needs.
- Provide simple explanations or definitions to help users understand concepts or terms.

1.4. Scope and Limitation:

A chatbot's scope includes providing automated customer support, task automation, information retrieval, personalized recommendations, user engagement, and integration with systems. Here are some scopes of chatbot:

- **Information Retrieval**: The chatbot can access and retrieve information from databases, knowledge bases, or external sources to provide users with accurate and relevant information in real-time. This enhances the user experience by delivering timely and helpful responses.
- **Automated Customer Support**: The chatbot can provide automated assistance to customers by answering common questions, resolving basic issues, and guiding users through simple tasks. This helps businesses offer round-the-clock support and reduce the workload on human agents.

Chatbots may face limitations in handling complex queries, understanding language nuances, maintaining contextual understanding, relying on predefined responses, addressing privacy and security concerns, lacking emotional intelligence, requiring ongoing training and maintenance, and being dependent on communication channels. Here are some limitations of chatbot:

- Complex Queries: The chatbot may struggle to handle complex or ambiguous queries that require human-like reasoning or contextual understanding beyond its programmed capabilities.
- Language and Linguistic Nuances: Limitations in natural language processing (NLP) may lead to misinterpretation of colloquial language, slang, or language variations, impacting the accuracy of responses.
- Limited Contextual Understanding: The chatbot may have difficulty maintaining context across multiple interactions or understanding implicit cues, leading to disjointed conversations and user frustration.

1.5. Development Methodology:

The development methodology of a chatbot typically involves several stages, similar to software development processes. Here's an overview of a common approach:

1. Define Objectives and Requirements: Clearly define the objectives of the chatbot project and identify the target audience. Gather requirements by understanding user needs, business goals, and technical constraints.
2. Design and Planning: Create a design plan outlining the chatbot's features, functionality, and user interface. Determine the technology stack, platform(s), and integration points required for development.
3. Development: Implement the chatbot using appropriate programming languages, frameworks, and tools. Develop natural language understanding (NLU) and natural language generation (NLG) components for processing user inputs and generating responses.
4. Testing: Conduct thorough testing to ensure the chatbot functions as intended and meets quality standards. Test for accuracy in understanding user inputs, correctness of responses, and overall system reliability.
5. Deployment: Deploy the chatbot on the intended platforms or channels, such as websites, messaging apps, or voice assistants. Configure hosting infrastructure and ensure scalability, availability, and security of the deployed system. Monitor performance metrics and user feedback post-deployment to address any issues or enhancements.
6. Training and Optimization: Continuously train the chatbot using real user interactions and feedback to improve its language understanding and response accuracy. Optimize algorithms, models, and dialogue flows based on performance metrics and user behavior data.
7. Maintenance and Updates: Regularly maintain and update the chatbot to address evolving user needs, business requirements, and technological advancements. Monitor system performance, address bugs or issues, and deploy patches or updates as needed.

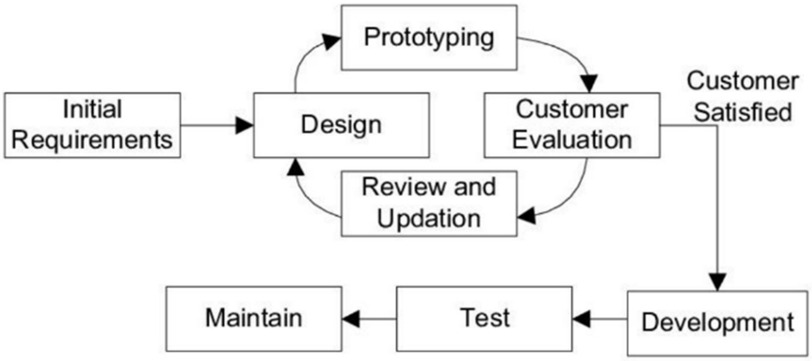


Figure-1: Development Methodology

1.6. Report Organization:

This report is divided into 6 chapters. Each chapter is divided into further sub-heading

- Chapter 1: Introduction

  It includes Introduction of chatbots, Problem Statement, Objectives of chatbots, Scope and Limitation, Development Methodology, Report Organization
- Chapter 2: Background Study and Literature Review

  Background Study of chatbots, Literature Review and similar work done under chatbot
- Chapter 3: System Analysis

  System Analysis of chatbots, Requirement Analysis which includes functional requirements and non-functional requirements, Feasibility Analysis (like technical, operational, economic and schedule feasibility analysis of chatbot), Data modeling using ER Diagrams Process modeling using DFD, Object modeling using Class and Object Diagrams, Dynamic modeling using State and Sequence Diagram, Process modeling using Activity Diagrams
- Chapter 4: System Design:

  Design, database design, forms and report design, interface and dialogue design, refinement of class, object, state, sequence and activity diagrams and algorithm details
- Chapter 5: Implementation and Testing

  Implementation which includes tools used (case tools, programming languages, database platforms, implementation details of modules (description of classes/procedures/functions/methods/algorithms), testing which includes test cases for unit testing & test cases for system testing and result analysis

- Chapter 6: Conclusion and Future Recommendation:
  Conclusion and future recommendations of chatbots

Chapter 2: Background Study and Literature Review

2.1. Background Study:

In Nepal, integrating chatbots holds significant potential for various sectors, especially in addressing language diversity and enhancing accessibility to services. With over 120 languages spoken, chatbots can bridge communication gaps by supporting multiple languages, improving citizen engagement in sectors like government services, healthcare, and tourism. They can provide information on local resources, weather updates, travel advisories, and cultural insights. Moreover, in remote areas with limited access to human support, chatbots can offer vital assistance in emergencies, education, and agricultural guidance. However, cultural sensitivities, literacy levels, and connectivity challenges must be considered for effective implementation in Nepal's diverse context.

In 1950, Alan Turing wondered if a computer program could talk to a group of people without realizing that their interlocutor was artificial. This question, named Turing test, is considered by many to be the generative idea of chatbots (Turing, 1950). The first chatbot with ELIZA name was constructed in 1966. ELIZA simulated a psychotherapist's operation, returning the user's sentences in the interrogative form Weizenbaum (1966). Its ability to communicate was limited, but it was a source of inspiration for the subsequent development of other chatbots (Klopfenstein et al., 2017).

Siri developed by Apple in 2010, pioneered the way for personal assistants. Users make inquiries and conversations with it through Messengers using voice commands, and it includes integration with audio, video, and image files. Siri makes recommendations and responds to user requests using various internet services, while it adapts, with constant use, to the users' language usages, searches, and desires.

2.2. Literature Review:

Pedro Antonio Tamayo , Ana Herrero , Javier Martín ,Carolina Navarro & José Manuel Tránchez (2020) report that "within the process of progressive digitization of materials and tools for teaching and distance learning of a subject of introduction to Microeconomics (quarterly, in year three of the Degree in Social Work), taught by the authors at the National

University of Distance Education (UNED), a virtual assistant in the form of chatbot, or conversational robot, called EconBot, has been designed and made available to students from 2017." Their paper "presents the reasons that led to its adoption, the process of its development, differentiating two phases, its characteristics and functions, the assessment of its usefulness and the role of teachers in the implementation of this type of technological innovation."

Kumar (2021) discusses educational chatbots (ECs) designed for pedagogical purposes. "These chatbots are strategized to provide personalized learning through the concept of a virtual assistant that replicates humanized conversation. Nevertheless, in the education paradigm, ECs are still novel with challenges in facilitating, deploying, designing, and integrating it as an effective pedagogical tool across multiple fields, and one such area is project-based learning." The chatbots were found to improve learning performance and teamwork and to facilitate collaboration among team members. "Nevertheless, affective-motivational learning outcomes such as perception of learning, need for cognition, motivation, and creative self-efficacy were not influenced by ECs.".

Chapter 3: System Analysis

3.1. System Analysis:

A system analysis of a chatbot involves examining its various components, functionalities, and interactions within the broader system it operates in. By conducting a comprehensive system analysis of a chatbot, stakeholders can gain insights into its functionality, architecture, usability, and performance, leading to informed decisions for optimization, enhancement, and future development efforts.

3.1.1. Requirement Analysis:

To obtain all the required information for the system development, interview, document review and observation were conducted. The system analysis, modeling deals with analyzing the proposed system. It includes the system use case diagram, ER diagram, context diagram, data flow diagram and their descriptions. After identifying the actors and use cases, the use cases are developed and textual descriptions are stated.

i. Functional Requirements:

Functional requirements of chatbots outline the specific capabilities and features that the chatbot should possess to fulfill its intended purpose and meet the needs of users and stakeholders. Here are some common functional requirements for chatbots:

- Natural Language Understanding (NLU): Interpret user input in natural language.
- Conversation Management: Guide users through conversation paths.
- Information Retrieval: Provide accurate and relevant information.
- Personalization: Customize responses based on user preferences.
- Feedback and Analytics: Gather user feedback to improve performance.

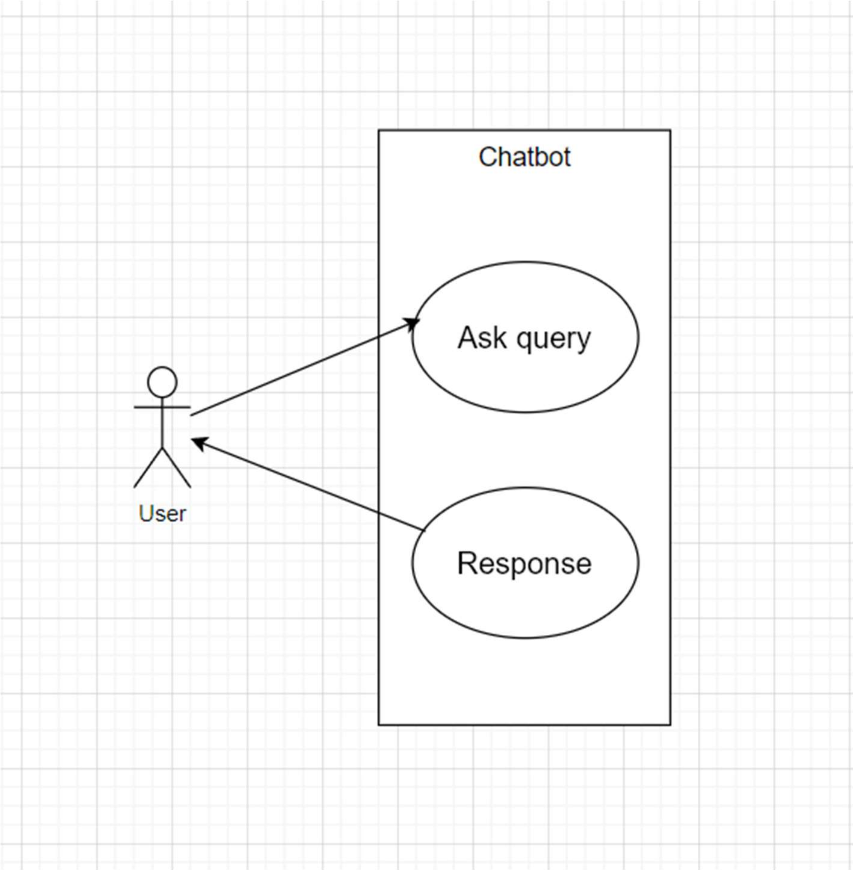


Figure-2: Use- Case Diagram

ii. Non- Functional Requirements:

These requirements focus on how the system works or how the system should behave by providing its quality attributes. These requirements include:

- Performance: Ensure fast response times to maintain user engagement. Ability to handle increasing user loads without degradation in performance.
- Usability: Chatbot interface should be user-friendly and easy to navigate. Ensure accessibility features for users with disabilities.
- Compatibility: Ensure compatibility with various devices and platforms. Support for different web browsers and versions.
- Integration with Existing Systems: Ensure compatibility and seamless integration with existing IT infrastructure and systems. Support for various APIs and standards used in integration.

These non-functional requirements are crucial for ensuring that a chatbot not only performs its intended functions effectively but also meets the broader expectations for reliability, security, usability, and compliance with relevant regulations and standards.

3.2. Planning:

3.2.1. Feasibility Analysis:

Feasibility analysis involves assessing the viability and practicality of implementing a chatbot project. Here's how feasibility analysis can be conducted across technical, operational, economic and scheduling dimensions:

i. Technical Feasibility:

- Evaluate the technical capabilities and requirements necessary to develop and deploy the chatbot.
- Assess the availability of required technologies, tools, and platforms for building the chatbot.
- Determine if the project aligns with the organization's existing technical infrastructure and architecture.
- Identify any technical challenges or limitations that may impact the development or integration of the chatbot.

ii. Operational Feasibility:

- Analyze how the chatbot will fit into existing business operations and workflows.
- Assess the readiness of personnel to adopt and support the chatbot within the organization.
- Evaluate potential changes to processes or roles needed to accommodate the chatbot's implementation.

iii. Economic Feasibility:

- Conduct a cost-benefit analysis to determine the financial viability of the chatbot project.
- Estimate the initial investment required for development, deployment, and maintenance of the chatbot.

- Identify potential cost savings or revenue opportunities associated with implementing the chatbot.
- Consider factors such as licensing fees, infrastructure costs, development resources, and ongoing support expenses.

iv. Schedule Feasibility:

- Evaluate the timeline and milestones for developing, testing, and deploying the chatbot.
- Assess the availability of resources, including personnel, technology, and infrastructure, to meet project deadlines.
- Identify potential bottlenecks or constraints that may impact the project schedule.
- Consider dependencies on external factors, such as integration with other systems or regulatory approvals.
- Develop a realistic project schedule with achievable timelines and contingency plans for potential delays or setbacks.

3.2.2. System Requirements:

- Hardware Requirement:
  Processor: Pentium V or higher
  RAM: 512 MB minimum
  Hard Disk: 5 GB

- Software Requirement
  FRONTEND: Html, CSS, node JS
  BACK END: Python
  DATABASE: -
  TOOLS: Xammp and Pytorch
  Drawing: Draw.io for E-R/Use Case diagram.

3.2.3. Work Schedule (Gantt Chart)

This project will comprise all the activities involved in SDLC. All these activities have been summarized in a Gantt chart below:

Table 1: Work Scheduling

S.N	Work	Time(date)
1.	System Analysis & Research	Dec 11- Dec 26
2.	Familiarization of Tools	Dec 27- Jan 2
3.	System Design	Jan 2- Jan 9
4.	System Development/ Coding	Jan 10- Feb 1
5.	System Testing & Debugging	Jan 25- Feb 15
6.	Documentation & Reporting	Dec 20- Feb 22
	Total Completion Time	73 Days

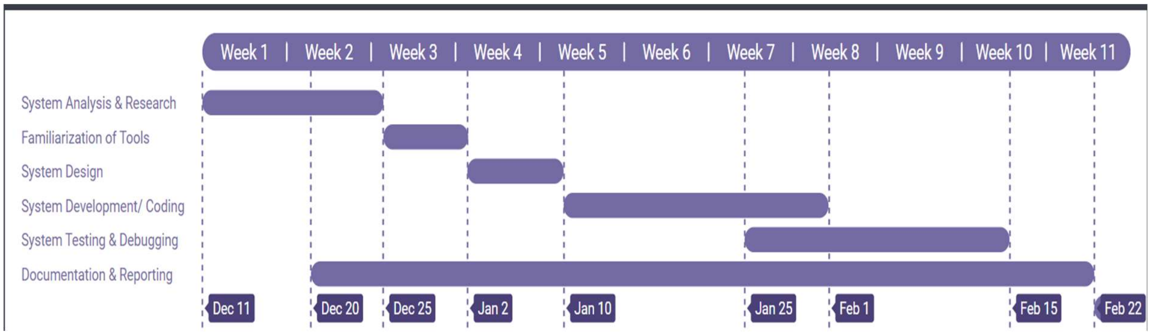


Figure-3: Gantt Chart

3.2.4. Data Flow Diagram:

1) Context Level Diagram:

A context-level diagram, also known as a level 0 DFD (Data Flow Diagram), provides an overview of the entire system, showing its boundaries and the interactions between it and external entities. Here's a context-level diagram for a chatbot system:

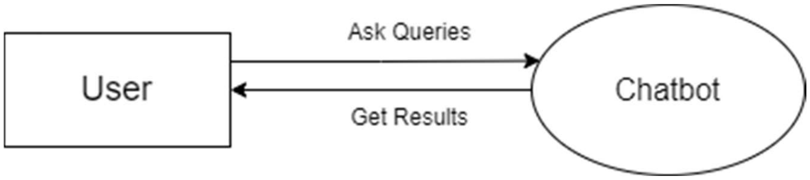


Figure-4 : Context Level Diagram for Chatbot

2) Level 1 DFD:

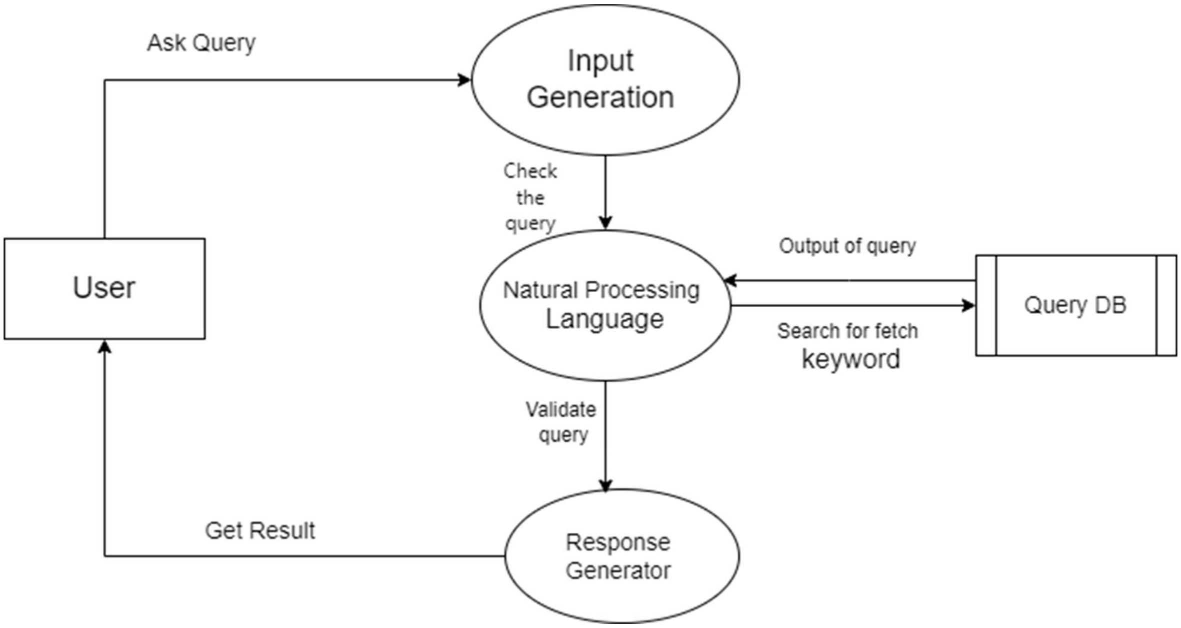


Figure-5: Level 1 DFD for chatbot

Chapter 4: System Design

4.1. Architecture of Chatbot:

An architecture of Chatbot requires a candidate response generator and response selector to give the response to the user's queries through text, images, and voice. The architecture of the Chatbot is shown in the below figure.

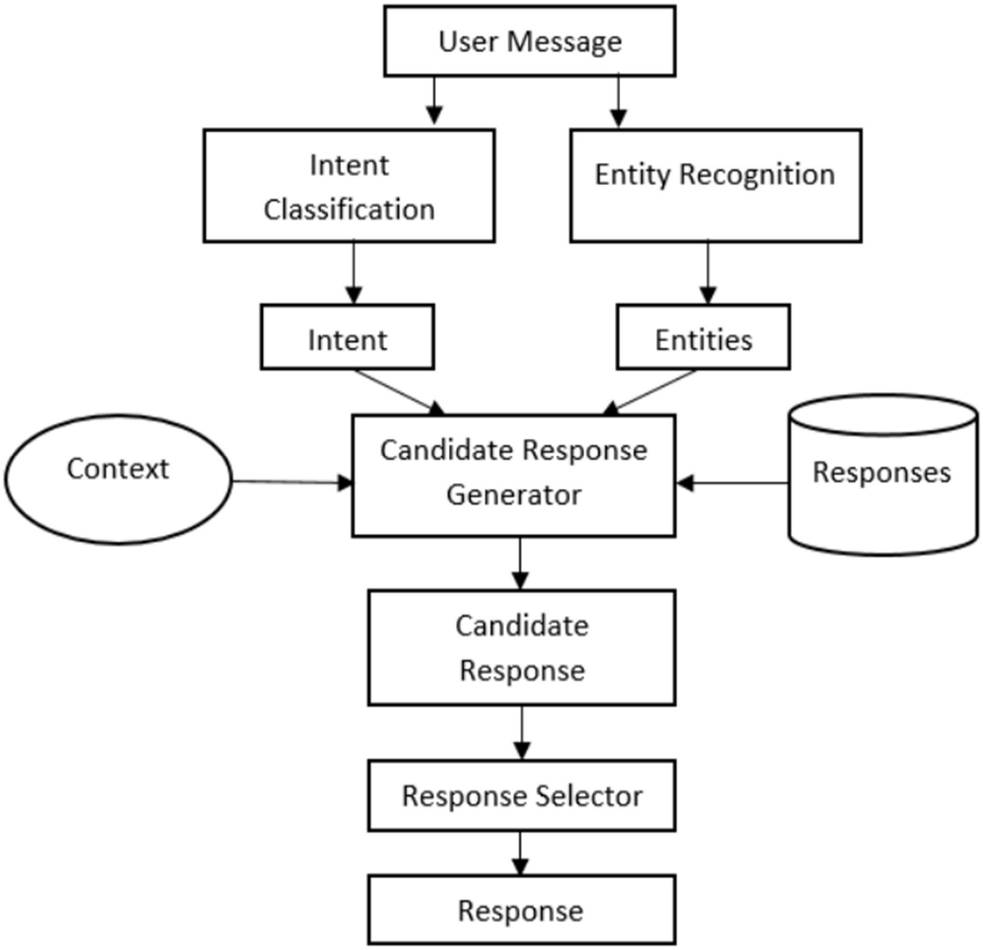


Figure-6: Work-Flow Diagram

In the above figure, user messages are given to an intent classification and entity recognition.

1. Intent: An intent in the above figure is defined as a user's intention, example the intent of the word "Good Bye" is to end the conversation similarly, the intent of the

word “What are some good Chinese restaurants” the intent would be to find a restaurant.

2. Entity: An entity in the Chatbot is used to modifies an intent and there are three types of entities they are system entity, developer entity and session entity.
3. Candidate Response Generator: The candidate response generator in the Chatbot do the calculations using different algorithms to process the user request. Then the result of these calculations is the candidate’s response.
4. Response Selector: The response selector in the Chatbot used to select the word or text according to the user queries to give a response to the users which should work better.

4.1. Activity Diagram:

An activity diagram is a type of UML (Unified Modeling Language) diagram that visually represents the flow of activities or actions within a system or process. It is commonly used to model the behavior of systems, software, or business processes. Activity diagrams are particularly useful for visualizing the steps involved in a workflow, including decisions, parallel activities, loops, and synchronization points.

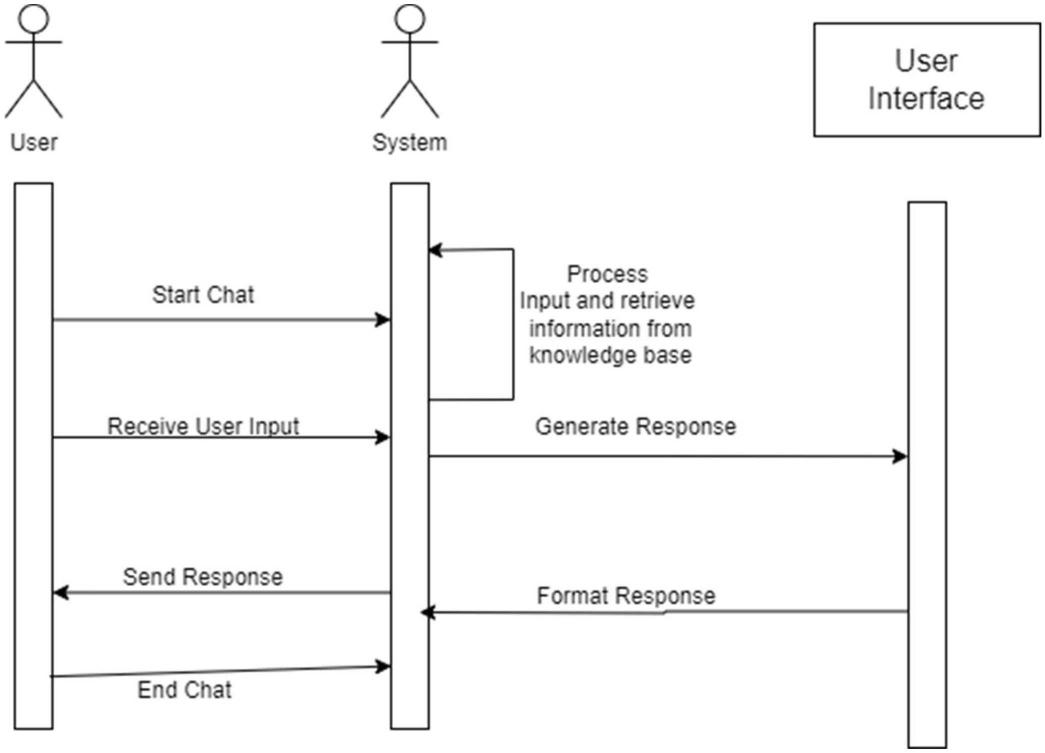


Figure-7: Activity Flow Diagram

4.2. Class Diagram:

A class diagram is a type of static structure diagram in the Unified Modeling Language (UML) that represents the structure and relationships of classes and other elements within a system. It provides a visual depiction of the classes in a system, their attributes, methods, and associations with other classes.

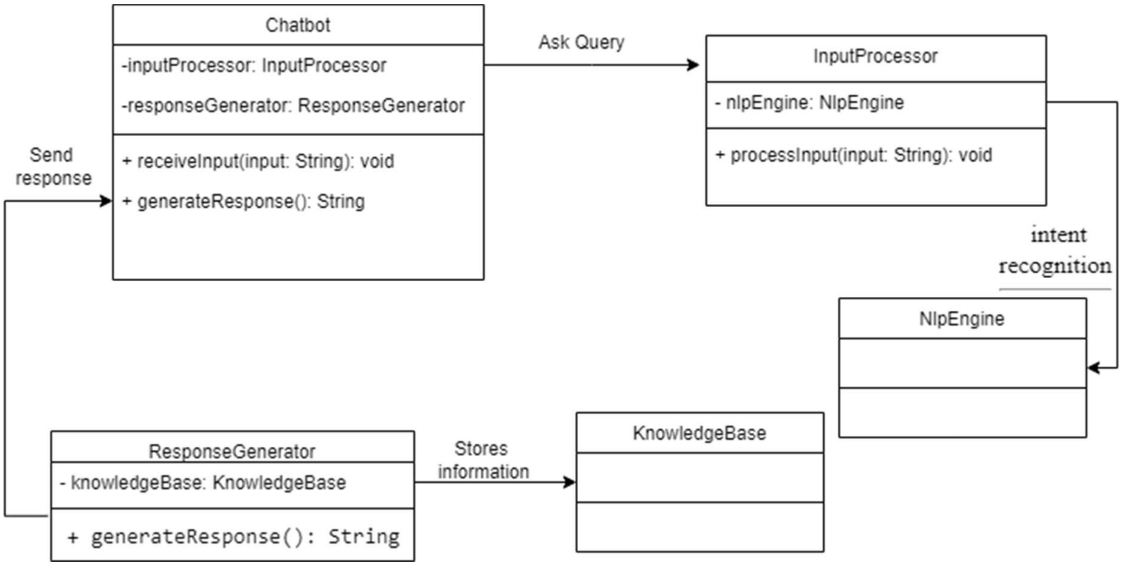

Chatbot
-inputProcessor: InputProcessor
-responseGenerator: ResponseGenerator
+ receiveInput(input: String): void
+ generateResponse(): String
Ask Query
InputProcessor
- nlpEngine: NlpEngine
+ processInput(input: String): void
intent recognition
NlpEngine
Send response
ResponseGenerator
- knowledgeBase: KnowledgeBase
+ generateResponse(): String
Stores information
KnowledgeBase

Figure-8: Class Diagram of chatbot

4.3. Algorithm Details:

4.3.1. Natural Language Processing:

Natural Language Processing (NLP) is a subfield of artificial intelligence (AI) and computational linguistics concerned with the interaction between computers and human (natural) languages. It focuses on enabling computers to understand, interpret, and generate human language in a way that is both meaningful and contextually relevant.

Some of the Natural Language Processing steps are:

- Text Classification: Categorizing text documents into predefined categories or classes, such as spam detection, sentiment analysis, and topic classification.

- Named Entity Recognition (NER): Identifying and extracting entities mentioned in text, such as people's names, organizations, locations, dates, and other important information.
- Intent analysis: the process of determining the underlying purpose or objective behind a communication or action, particularly in the context of natural language understanding and artificial intelligence.
- Question Answering (QA): Automatically generating answers to questions posed in natural language, by retrieving relevant information from a knowledge base or corpus of text.

4.3.2. Feed-Forward Network

Feed-Forward Neural Network is a classification algorithm. It is an artificial neural network in which the nodes don't form any cycle or directed acyclic graph. It is the simplest form of neural network. Its opposite is a recurrent neural network in which the nodes form a cycle. It forms a single-layer perceptron. The structure of a feedforward is an advantage for certain applications. Examples of the feed-forward neural network are radial basis function networks, which uses activation function.

How does a Feed-Forward Network work?

It is a single-layer perceptron. In this model series of input enters the layer and is then multiplied with its weights. All the values are added to get the sum of weighted inputs and check if the value is above a specific threshold. If it is then it is set to zero and if less than the threshold then it is set to -1. This model is mostly used in classification tasks. This process of training and learning produces a form of gradient descent. Using a property known as the delta rule, the neural network can compare the outputs of its nodes with the intended values, thus allowing the network to adjust its weights through training to produce more accurate output values. Feed-Forward neural networks have a very simple network structure. Hence to improve the chatbot we can use complex models such as RNN (seq2seq model) or Bidirectional Recurrent Neural Network (BRNN). The BRNN was chosen, like conversation or input to the Chatbot is dynamic, which means the length of the input is unfixed.

4.3.3. Feed-Forward Network Algorithm worked in our system

In a chatbot system based on a feedforward neural network (FNN) algorithm, the chatbot processes user inputs through layers of interconnected neurons without cycles (i.e., feedforward). Here's how it typically worked in our Chatbot system:

Step 1: Input Layer: User input, usually in the form of text, is fed into the input layer of the neural network.

Step 2: Hidden Layers: The input data is then processed through one or more hidden layers, where each layer consists of multiple neurons. Each neuron applies weights to the input data and passes it through an activation function to introduce non-linearity. These hidden layers help the network learn complex patterns and representations from the input data.

Step 3: Output Layer: The processed data from the hidden layers is finally fed into the output layer, where it produces the chatbot's response. In a chatbot system, the output layer typically generates text responses to the user's input.

Step 4: Training: Before deployment, the neural network is trained on a large dataset of conversational data. During training, the network adjusts its internal parameters (weights and biases) based on the input-output pairs provided in the training data, using techniques like backpropagation and gradient descent to minimize the prediction error.

Step 5: Prediction: Once trained, the chatbot can predict responses to new user inputs by passing them through the trained neural network. The network computes the output based on the learned patterns and relationships encoded in its parameters.

Step 6: Feedback Loop: In some cases, the chatbot may incorporate a feedback loop where user interactions are used to further train or fine-tune the neural network over time, improving its performance and responsiveness.

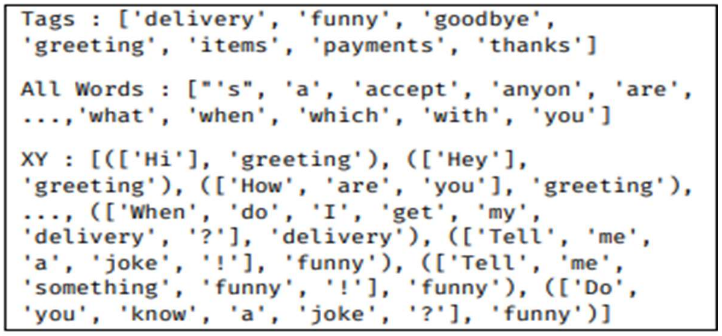
Overall, in a feedforward neural network-based chatbot system, the network learns to map user inputs to appropriate responses through layers of computation, enabling it to engage in human-like conversations.

Chapter-5: Implementation And Testing

5.1. Implementation:

We tokenized and stemmed the words after loading the JSON file containing roughly 2800 intents and their associated tags, patterns, and answers. Tokenization is the process of extracting an array of distinct words from a phrase, whereas stemming is the process of reducing a word to its word stem, which affixes to suffixes and prefixes or to the roots of words known as lemma. The stem 'tak' will match words like 'take,' 'taking,' 'takers,' and so on. We could tidy up the words list and eliminate any duplicates, but this will be enough for now.

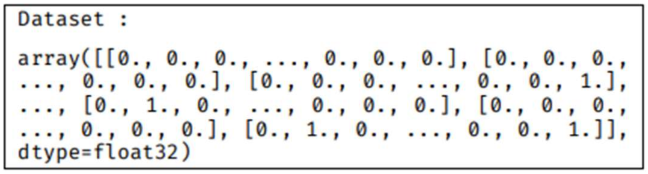
The loaded data will be converted into an array of words, an array of tags, and an array of sets where words with their respective tags will be generated.



```
Tags : ['delivery', 'funny', 'goodbye',
'greeting', 'items', 'payments', 'thanks']

All Words : ["'s", 'a', 'accept', 'anyon', 'are',
...,'what', 'when', 'which', 'with', 'you']

XY : [(['Hi'], 'greeting'), (['Hey'],
'greeting'), (['How', 'are', 'you'], 'greeting'),
..., (['When', 'do', 'I', 'get', 'my',
'delivery', '?'], 'delivery'), (['Tell', 'me',
'a', 'joke', '!'], 'funny'), (['Tell', 'me',
'something', 'funny', '!'], 'funny'), (['Do',
'you', 'know', 'a', 'joke', '?'], 'funny')]
```

Figure-9: Tokenization of array

Unfortunately, this data structure will not function with PyTorch, we must convert it from documents of words to an array of integers. The bag of words function is used to perform this conversion. The dataset is now ready for training and looks like this:



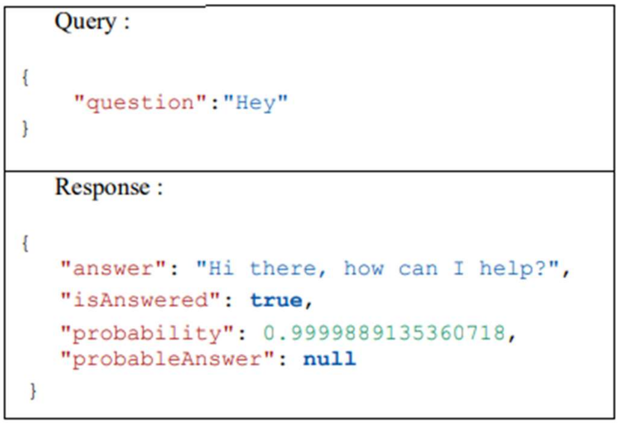
```
Dataset :

array([[0., 0., 0., ..., 0., 0., 0.], [0., 0., 0.,
..., 0., 0., 0.], [0., 0., 0., ..., 0., 0., 1.],
..., [0., 1., 0., ..., 0., 0., 0.], [0., 0., 0.,
..., 0., 0., 0.], [0., 1., 0., ..., 0., 0., 1.]],
dtype=float32)
```

Figure-10: Dataset of Chatbot

The dataset presented below is used to load the data into PyTorch, with batch size and shuffle specified in the hyperparameters. The PyTorch-loaded data is processed through the

Neural Net Class once more, with the input size, hidden size, and output size supplied as hyperparameters. The size of the input array will equal the size of the training array, and the size of the output array will be the number of tags. As a result, the model can categorize a phrase or an array of words based on their probabilities. The activation function ReLU (Rectified Linear Unit) is used to turn the model's output into tags with probability. The Cross-Entropy Loss is assessed during model training, and the Adam Optimizer is applied to the model parameters. The model is then trained for 200 epochs, with each step dumping the model state, input size, hidden size, output size, and all words and tags into a "data.pth" file to be utilized later. For deployment, the file with the lowest Cross-Entropy Loss is utilized. Before putting it into the model, the user-asked query is tokenized, stemmed, and turned into a bag of words. The model answers with the corresponding tags and their level of confidence. The response is chosen based on the tags with the highest level of confidence.

The model is hosted on Heroku Server and has a REST API. To receive a response, a post request containing a question must be submitted.



Query :

```
{
   "question":"Hey"
}
```

Response :

```
{
  "answer": "Hi there, how can I help?",
  "isAnswered": true,
  "probability": 0.9999889135360718,
  "probableAnswer": null
 }
```

Figure-11: Query & Answer

5.2. Testing:

The testing is done to validate and verify the chatbot. Testing of a chatbot involves assessing its functionality, usability, performance, and reliability to ensure it meets the intended requirements and provides a satisfactory user experience.

5.2.1 Unit Testing:

In chatbot system unit testing is done by testing every individual component of the chatbot system and we got the result as follows:

Table-2: Cases for Chat Module Testing

S.N	Test Case	Input	Outcome	Pass/ Fail
1.	Chatbot	User entered greetings	Greeting replied to the user.	Pass
2.	Chatbot	User tapped random question	Random question generated.	Pass
3.	Chatbot	User send submit	Answer replied to the user.	Pass
4.	Chatbot	User send empty submit	Message is required	Pass
5.	Chatbot	Drag moon icon to down	Interface in dark mode	Pass
6.	Chatbot	Drag sun icon to down	Interface in light mode	Pass

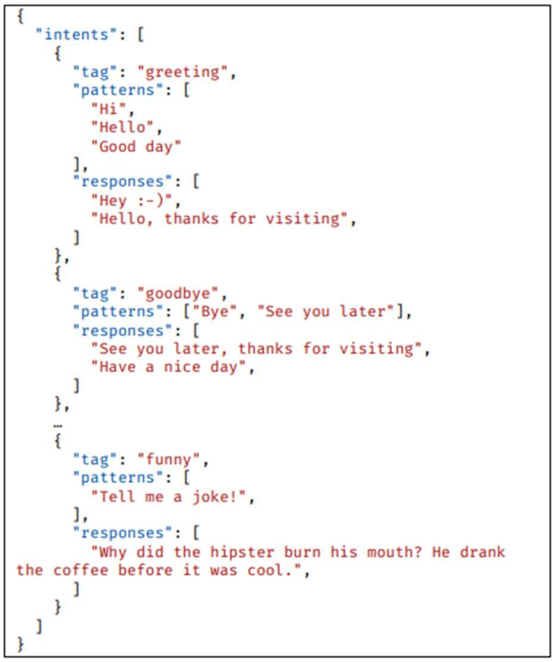
5.2.2. System Testing

System Testing is focused on assessing the system’s reliability. It helps to determine the optimality of the internal structure and the outputs generated by the system meets the system requirements. Faults that are discovered during systems testing are passed back to the development phase for repair. Then the faults are recovered and then the system is tested again as a whole.

In chatbot system, system testing is also done after integration testing in order to ensure that the whole system functions properly. After the integration testing, the whole system working process was checked. We found that the output was as per the system specifications and hence the system was found to work properly.

5.3. Training Datasets:

The dataset is made up of WikiQA and smaller query response datasets. It has around 2800 intents, each with its own unique tag and a set of patterns (queries) and replies (answers). The dataset's tags enable the model to recognize trends and responses. The pattern specifies the sort of question that can be asked in response to the same query. Because many questions might have the same meaning but different phrasing, the collection of these questions is recorded in patterns. The answer is similar in that it holds many responses with the same broad meaning but different terms.



```
{
  "intents": [
    {
      "tag": "greeting",
      "patterns": [
        "Hi",
        "Hello",
        "Good day"
      ],
      "responses": [
        "Hey :-)",
        "Hello, thanks for visiting",
      ]
    },
    {
      "tag": "goodbye",
      "patterns": ["Bye", "See you later"],
      "responses": [
        "See you later, thanks for visiting",
        "Have a nice day",
      ]
    },
    …
    {
      "tag": "funny",
      "patterns": [
        "Tell me a joke!",
      ],
      "responses": [
        "Why did the hipster burn his mouth? He drank
the coffee before it was cool.",
      ]
    }
  ]
}
```

Figure-12: Training Dataset

5.4. Testing Dataset:

Creating a testing dataset for a chatbot involves generating various scenarios and user inputs to evaluate the performance of the chatbot. Here's an example of a testing dataset for a chatbot designed to provide information and recommendation:

Table-3: Testing Dataset

Scenario	User Input	Response
User want to know about recipe of white chocolate	what is white chocolate made of	It commonly consists of cocoa butter , sugar , milk solids and salt
User want to know about kidney	where are the kidneys in your body	Each kidney excretes urine into a ureter , itself a paired structure that empties into the urinary bladder .
User want to know about direct-Marketing	what is direct marketing channel	A well-executed direct advertising campaign can prove a positive return on investment by showing how many potential customers responded to a clear call-to-action.
User want to know about goodFriday	What is the significance of Good Friday?	Based on the details of the Canonical gospels , the Crucifixion of Jesus was most likely to have been on a Friday (the day before the Sabbath) ().

Chapter-6: Conclusion & Future Recommendation

6.1. Conclusion:

Chatbots are programs built to automatically engage with received messages. Chatbots can be programmed to respond the same way each time, to respond differently to messages containing certain keywords and even to use machine learning to adapt their responses to fit the situation. For many applications, the chatbot is connected to the database. The database is utilized to sustain the chatbot and provide appropriate responses to every user. NLP can translate human language into data information with a blend of text and patterns that can be useful to discover applicable responses.

There are NLP applications, programming interfaces, and services that are utilized to develop chatbots. And make it possible for all sorts of businesses – small, medium or largescale industries. The primary point here is that smart bots can help increase the customer base by enhancing the customer support services, thereby helping to increase sales. AI-powered chatbots understand free language, but also have a predefined flow to make sure they solve the user's problem. They can remember the context of the conversation and the user’s preferences.

These chatbots can jump from one point of conversation scenario to another when needed and address random user requests at any moment. These chatbots use Machine Learning, AI and Natural Language Processing (NLP) to understand people.

The goal of NLP is to make the interaction between computers and humans feel like communication between two people. With the help of NLP people can freely interact with chatbots asking a question.

6.2 Future Work:

The future of chatbots holds exciting possibilities for further development and innovation. Here are some potential areas of future work for chatbots:

1) Advanced Natural Language Understanding (NLU): Improving the accuracy and sophistication of NLU models to better understand user intent, context, and sentiment.

Incorporating more advanced linguistic features and semantics into NLU systems for nuanced understanding of user input.

2) Domain-Specific and Expert Chatbots: Developing chatbots tailored to specific industries or domains, such as healthcare, finance, education, or customer service, with specialized knowledge and expertise. Creating expert chatbots that can provide expert-level advice, recommendations, and insights in specific fields.
3) Human-Agent Collaboration: Facilitating seamless collaboration between humans and chatbots in various domains, enabling chatbots to assist humans in tasks, decision-making, and problem-solving. Designing chatbots as collaborative partners that complement human capabilities and enhance productivity and efficiency.
4) Continuous Improvement and Feedback Loop: Establishing feedback mechanisms and analytics to monitor chatbot performance, gather user feedback, and iteratively improve the system over time. Adopting agile development practices and iterative design methodologies to rapidly iterate and evolve chatbot capabilities in response to changing user needs and market dynamics.
5) Personalization and Context Awareness: Developing chatbots that can personalize responses based on user preferences, history, and behavior. Enhancing context awareness to maintain continuity in conversations across different channels and sessions.

The future of chatbots lies in pushing the boundaries of AI, natural language processing, and human-computer interaction to create more intelligent, intuitive, and empathetic conversational agents that enrich our digital experiences and improve our everyday lives.

REFERENCES

[1] S., V., & S. (n.d.). Open Data Chatbot.

[2] P., R., ., & A. (n.d.). Feedback-Based Self-Learning In Large-Scale Conversational AI Agents.

[3] Chatbot Tutorial — PyTorch Tutorials 1.10.0+cu102 Documentation. (n.d.). Chatbot Tutorial — PyTorch Tutorials 1.10.0+cu102 documentation. https://pytorch.org/tutorials/beginner/chatbot_tutorial.html.

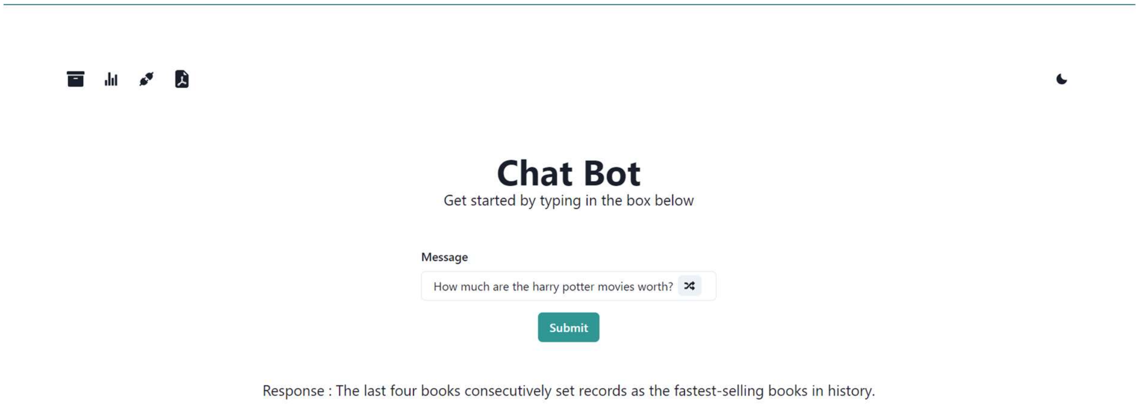
[4] Chat Bot With PyTorch - NLP And Deep Learning | Python Engineer. (2020, June 14). Chat Bot With PyTorch - NLP And Deep Learning | Python Engineer. https://www.python-engineer.com/posts/chatbotpytorch/.

[5] Knuth G. (2017, September 9). Contextual Chatbots With Tensorflow | By Gk_| Chatbots Magazine. Medium. https://chatbotsmagazine.com/contextual-chat-bots-with-tensorflow4391749d0077.

[6] chatbots in education: a (short) literature review https://pontydysgu.eu/2022/12/chatbots-in-education-a-short-literature-review/

[7] An Intelligent Web App Chatbot https://ieeexplore.ieee.org/document/9276948

APPENDIX



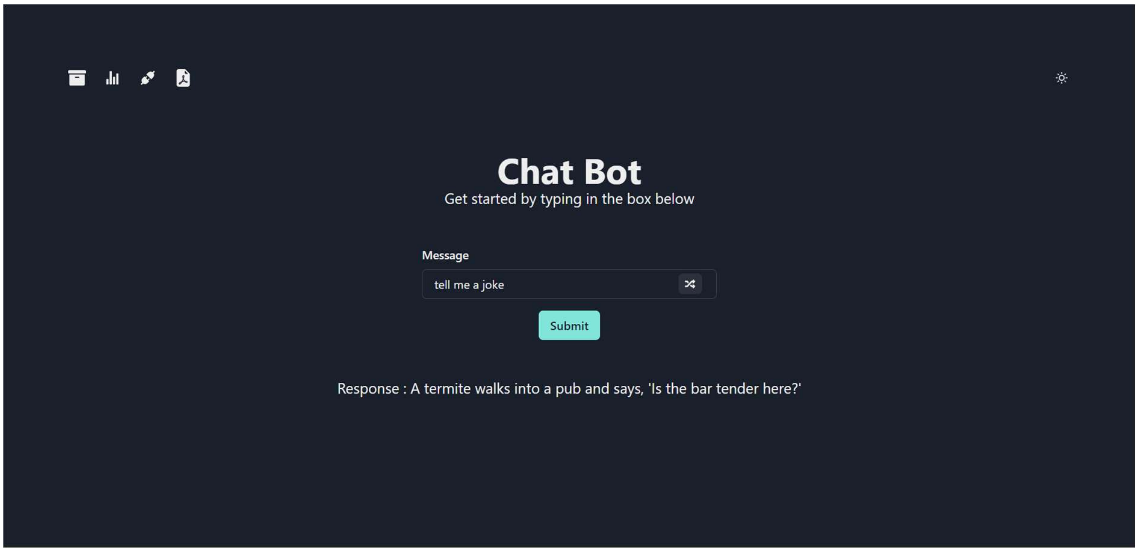
Chat Bot

Get started by typing in the box below

Message

How much are the harry potter movies worth?

Response : The last four books consecutively set records as the fastest-selling books in history.



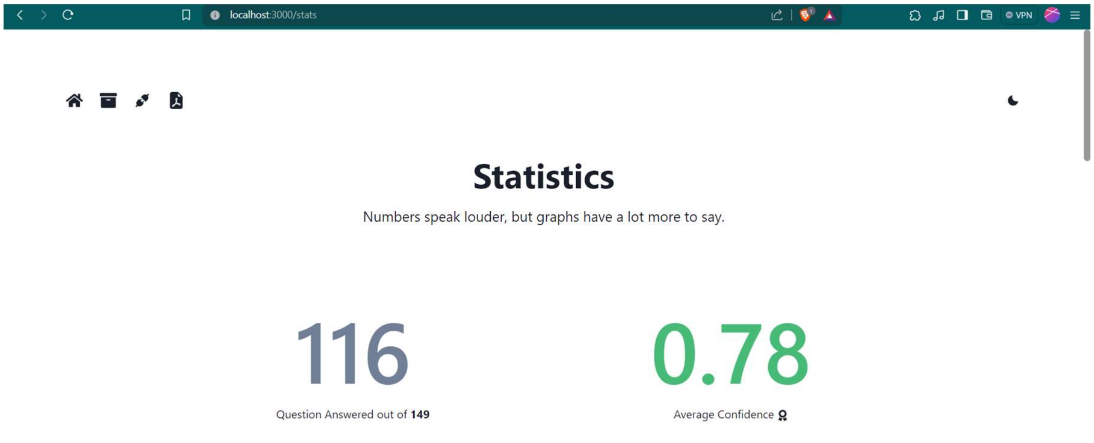
Chat Bot

Get started by typing in the box below

Message

tell me a joke

Response : A termite walks into a pub and says, 'Is the bar tender here?'



Statistics

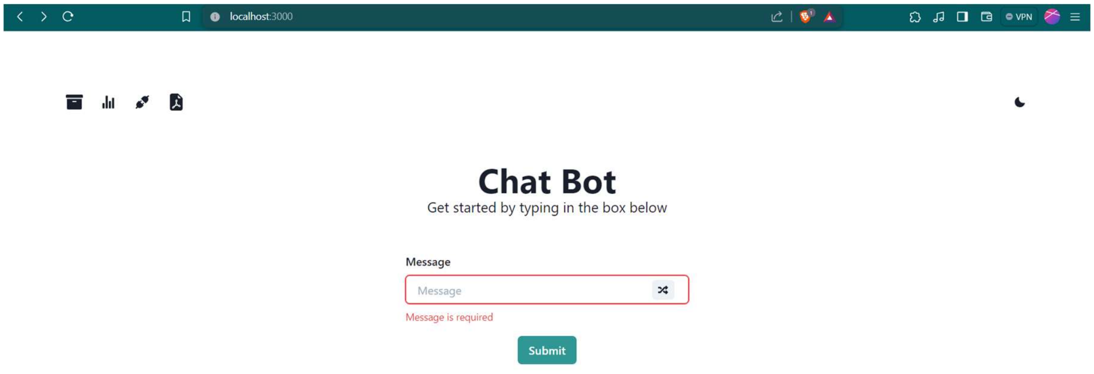
Numbers speak louder, but graphs have a lot more to say.

116

Question Answered out of **149**

0.78

Average Confidence



Chat Bot

Get started by typing in the box below

Message

Message

Message is required

Submit

Sagarmatha College of Science and Technology

Sanepa, Lalitpur

Supervisor Name: Sobit Thapa

PROJECT LOGBOOK

Date	Meeting Discussion	Signature	Remarks
Mangsir 21	Group Formation And Faculty selection		
Mangsir 28	Requirement Collection and Project name choose		
Poush 9	Planning on project and work division discussion		
Poush 17	Discussion on development methodology		
Poush 29	Frontend first look setup		
Magh 7	Backend tools deciding		
Magh 10	Discussion on Algorithm FNN implementation on project		
Magh 19	Backend and Frontend linking discussion		
Magh 23	First testing and error debugging regarding the project		
Falgun 05	Final Testing and final report documentation of project		