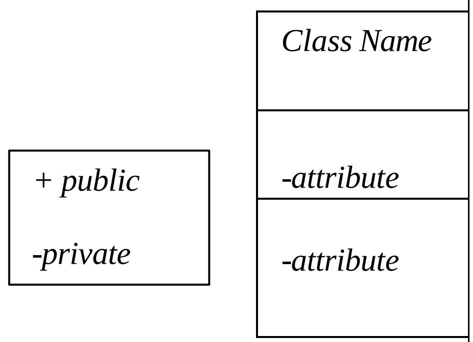
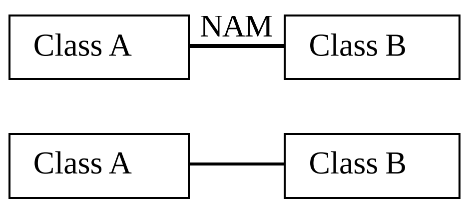
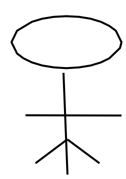
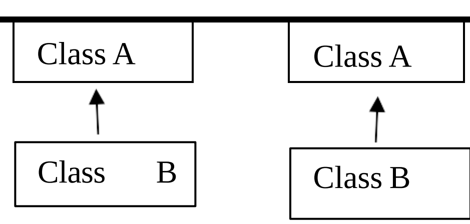
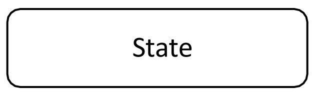
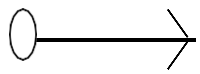
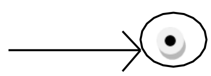
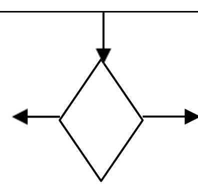
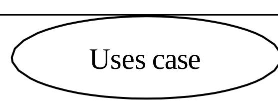
CREATING A DATASET FOR AN INDUSTRIAL ROBOT WIZARD-OF-OZ DIALOGUE

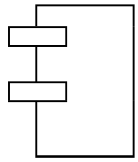
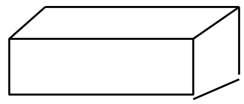
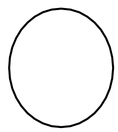
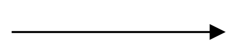
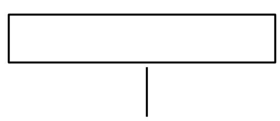
TABLE OF CONTENTS

LIST OF FIGURES

LIST OF SYSMBOLS

S.NO	NOTATION NAME	NOTATION	DESCRIPTION
1.	Class		Represents a collection of similar entities grouped together.
2.	Association		Associations represents static relationships between classes. Roles represents the way the two classes see each other.
3.	Actor		It aggregates several classes into a single classes.
4.	Aggregation		Interaction between the system and external environment

5.	Relation (uses)	uses	Used for additional process communication.
6.	Relation (extends)		Extends relationship is used when one use case is similar to another use case but does a bit more.
7.	Communication		Communication between various use cases.
8.	State		State of the processes.
9.	Initial State		Initial state of the object
10.	Final state		Final state of the object
11.	Control flow		Represents various control flow between the states.
12.	Decision box		Represents decision making process from a constraint
13.	Use case		Interact ion between the system and external environment.

14.	Component		Represents physical modules which are a collection of components.
15.	Node		Represents physical modules which are a collection of components.
16.	Data Process/State		A circle in DFD represents a state or process which has been triggered due to some event or action.
17.	External entity		Represents external entities such as keyboard, sensors, etc.
18.	Transition		Represents communication that occurs between processes.
19.	Object Lifeline		Represents the vertical dimensions that the object communications.
20.	Message		Represents the message exchanged.

ABSTRACT:

Enabling a flexible and natural human-robot interaction (HRI) for industrial robots is a critical yet challenging task that can be facilitated by the use of conversational artificial intelligence (AI). Prior research has concentrated on strengthening interactions through the deployment of social robots, while disregarding the capabilities required to boost the flexibility and user experience associated with human-robot collaboration (HRC) on manufacturing tasks. One of the main challenges is the lack of publicly available industrial-oriented dialogue datasets for the training of conversational AI. In this work, we present an Industrial Robot Wizard-of-Oz Dialoguing Dataset (IRWoZ) focused on enabling HRC in manufacturing tasks. The dataset covers four domains: assembly, transportation, position, and relocation. It is created using the Wizard-of-Oz technique to be less noisy. We manually constructed, annotated and validated dialogue segments (e.g., intentions, slots, annotations), as well as the responses. Building upon the proposed dataset, we benchmark it on the state-of-the-art (SoTA) language models, generative pretrained (GPT-2) models, on dialogue state tracking and response generation tasks.

CHAPTER-1

INTRODUCTION

In the pursuit of fostering a more adaptive and seamless interaction between humans and industrial robots in manufacturing environments, the incorporation of conversational artificial intelligence (AI) has emerged as a pivotal avenue. While previous research has primarily focused on enhancing interactions through the integration of social robots, there remains a critical gap in addressing the specific capabilities necessary to elevate flexibility and user experience within human-robot collaboration (HRC) for manufacturing tasks. A key obstacle in advancing this domain is the scarcity of publicly available dialogue datasets tailored for industrial contexts, hindering the training of effective conversational AI models. This work endeavors to bridge this gap by introducing the Industrial Robot Wizard-of-Oz Dialoguing Dataset (IRWoZ), meticulously crafted to empower HRC in manufacturing tasks. Encompassing four distinct domains—assembly, transportation, position, and relocation—the dataset is curated using the Wizard-of-Oz technique to mitigate noise and enhance authenticity. Through manual construction, annotation, and validation of dialogue segments, including intentions, slots, and annotations, as well as responses, the IRWoZ dataset serves as a valuable resource. Beyond its creation, the dataset is subjected to benchmarking against state-of-the-art language models, particularly generative pretrained models like GPT-2, evaluating its efficacy in tasks related to dialogue state tracking and response generation. This initiative not only addresses the critical need for specialized datasets but also contributes to the advancement of language models tailored for enhancing human-robot collaboration in industrial settings.

1.2 SCOPE OF THE PROJECT

The scope of the project revolves around advancing human-robot collaboration (HRC) in manufacturing by integrating conversational artificial intelligence (AI). The primary objective is to enhance flexibility and user experience in industrial settings, addressing a gap in previous research that predominantly focused on social robots. To bridge this void, the project introduces the Industrial Robot Wizard-of-Oz Dialoguing Dataset (IRWoZ). This meticulously crafted dataset spans four domains—assembly, transportation, position, and relocation—and is curated using the Wizard-of-Oz technique to ensure authenticity and reduce noise. Through manual construction, annotation, and validation, the dataset encompasses dialogue segments, intentions, slots, annotations, and responses, positioning itself as a valuable resource for HRC. The project also includes benchmarking the IRWoZ dataset against state-of-the-art language models, particularly GPT-2, to assess its efficacy in dialogue state tracking and response generation tasks. Beyond filling the scarcity of publicly available dialogue datasets tailored for industrial contexts, this initiative contributes significantly to the advancement of language models tailored for enhancing human-robot collaboration in manufacturing environments.

1.3 OBJECTIVE

This research endeavors to advance the field of human-robot collaboration (HRC) in manufacturing tasks through the strategic incorporation of conversational artificial intelligence (AI). While existing studies have predominantly emphasized the integration of social robots, our work identifies a critical gap in the exploration of capabilities essential for enhancing flexibility and user experience within HRC for manufacturing. To address this, we introduce the Industrial Robot Wizard-of-Oz Dialoguing Dataset (IRWoZ), a purpose-built dataset tailored for industrial contexts, covering diverse domains such as assembly, transportation, position, and relocation. Utilizing the Wizard-of-Oz technique during its creation, we aim to reduce noise and enhance the authenticity of the dataset. The manual construction, annotation, and validation of dialogue segments, including intentions, slots, annotations, and responses, underscore our commitment to ensuring the dataset's quality and accuracy. Additionally, we conduct a benchmarking exercise by evaluating the IRWoZ dataset against state-of-the-art language models, particularly generative pretrained models like GPT-2. Through this comprehensive approach, our objectives encompass not only filling the void in industrial-oriented dialogue datasets but also providing a robust foundation for training conversational AI models specifically tailored to elevate human-robot collaboration in manufacturing environments.

1.4 EXISTING SYSTEM:

- Conversational machine reading comprehension (MRC) is a new question answering task, which is more challenging compared to traditional single-turn MRC since it requires a better understanding of conversation history.
- In this paper, a novel neural network model for conversational reading comprehension.

1.4.1 EXISTINGSYSTEM DISADVANTAGES:

- Failed to maintain consistency.
- Complexity is high

CHAPTER 2

LITERATURE REVIEW

2.1 LITERATURE SURVEY

TITLE: A Collaborative Robot Cell for Random Bin-Picking Based on Deep Learning Policies and a Multi-Gripper Switching Strategy.

AUTHORS: A. S. Olesen, B. B. Gergaly, E. A. Ryberg, M. R. Thomsen, and D. Chrysostomou.

YEAR: 2020.

DESCRIPTION:

This work presents a collaborative robot cell designed for random bin-picking in industrial settings. The approach employs deep learning policies and a multi-gripper switching strategy to enhance the robot's ability to handle diverse objects in a bin. The integration of deep learning allows the robot to adapt to varying shapes and sizes of objects, demonstrating a flexible and intelligent robotic system for industrial applications. The multi-gripper switching strategy further contributes to the efficiency and versatility of the system by enabling the robot to choose the most suitable gripper for a given object, optimizing the overall bin-picking process.

TITLE: A Dual-Arm Collaborative Robot System for the Smart Factories of the Future.

AUTHORS: J. F. Buhl, R. Grønhøj, J. K. Jørgensen, G. Mateus, D. Pinto, J. K. Sørensen, S. Bøgh, and D. Chrysostomou.

YEAR: 2019.

DESCRIPTION: This research introduces a dual-arm collaborative robot system tailored for smart factories. The system is designed to enhance efficiency and flexibility in manufacturing environments by employing dual arms for improved task handling. The collaborative nature of the robot system facilitates interaction with human workers, promoting a synergistic relationship between automation and human expertise. The work addresses the evolving needs of smart factories, aiming to contribute to the advancement of Industry 4.0 by integrating intelligent and adaptable robotic solutions into the manufacturing landscape.

TITLE: Plug & produce robot assistants as shared resources: A simulation approach.

AUTHORS: E. R. da Silva, C. Schou, S. Hjorth, F. Tryggvason, and M. S. Sørensen.

YEAR: 2022.

DESCRIPTION: This paper introduces a simulation-based approach to explore the concept of plug & produce robot assistants as shared resources. The research aims to assess the feasibility and benefits of utilizing robotic assistants in manufacturing through simulation. The findings contribute insights into the potential of collaborative robots as shared resources in industrial settings. The study involves a comprehensive exploration of the simulation results, aiming to provide valuable insights into the practicality and benefits of integrating robotic assistants as shared resources in industrial workflows. The term "plug & produce" suggests a seamless and easily deployable integration process, emphasizing the flexibility and adaptability of robotic assistants within the manufacturing domain.

TITLE: Integration and assessment of multiple mobile manipulators in a real-world industrial production facility.

AUTHORS: S. Bogh.

YEAR: 2014.

DESCRIPTION: Focusing on the integration and assessment of multiple mobile manipulators, this paper addresses their role in a real-world industrial production facility. The research evaluates the performance and practical implementation of mobile manipulators, providing valuable insights into their integration into industrial workflows. The simulation-based approach allows researchers to replicate real-world scenarios, enabling a detailed examination of the interactions and performances of robotic assistants in various manufacturing contexts. Through the simulation, the authors assess factors such as efficiency, resource utilization, and overall system dynamics, offering a nuanced understanding of the potential challenges and advantages associated with the deployment of collaborative robots in industrial settings.

2.2 PROPOSED SYSTEM

- The WoZ approach (i.e., human-to-human method) is used in this study to mimic dialogue between shop floor worker and industrial robots.
- To assist the process of collecting the dialogue corpus, a web application built on the Flask web framework is designed and implemented the overall architecture of the web application of the IRWoZ dialogue simulation framework is depicted.
- It follows the Client-Server style architecture. The client side comprises user and wizard interfaces, while the server side includes the dialogue controller, belief state verification, dialogue act generation, dialogue annotation, and dialogue auto saving.

2.2.1 PROPOSED SYSTEM ADVANTAGES:

- Time consumption is less
- They also have some limitations and challenges, such as the potential for high computational requirements.
- Maintains consistency.

CHAPTER 3

PROJECT DESCRIPTION

3.1 GENERAL:

This project aims to enhance the interaction between humans and industrial robots in manufacturing environments by incorporating conversational artificial intelligence (AI). While previous research has focused on social robots, there exists a gap in addressing specific capabilities crucial for improving flexibility and user experience in human-robot collaboration (HRC) for manufacturing tasks. The scarcity of publicly available dialogue datasets tailored for industrial contexts is identified as a key obstacle. To address this gap, the project introduces the Industrial Robot Wizard-of-Oz Dialoguing Dataset (IRWoZ), meticulously designed to empower HRC in manufacturing. Covering four distinct domains—assembly, transportation, position, and relocation—the dataset utilizes the Wizard-of-Oz technique to reduce noise and enhance authenticity. Through manual construction, annotation, and validation, including intentions, slots, annotations, and responses, the IRWoZ dataset serves as a valuable resource. Beyond its creation, the dataset is benchmarked against state-of-the-art language models like GPT-2, assessing its effectiveness in dialogue state tracking and response generation. This initiative not only meets the critical need for specialized datasets but also contributes to the advancement of language models tailored for improving human-robot collaboration in industrial settings.

3.2 METHODOLOGIES

3.2.1 MODULES NAME:

Modules Name:

1. Data Acquisition

2. Conversational AI Model Development

3. Wizard-of-Oz Technique Implementation

4. Benchmarking and Evaluation

5. User Interface (UI) Integration

3.2.2 MODULES EXPLANATION:

1. Data Acquisition Module:

- **Dataset Creation**: Develop a module dedicated to creating and curating the Industrial Robot Wizard-of-Oz Dialoguing Dataset (IRWoZ). This involves designing processes for manual construction, annotation, and validation of dialogue segments across diverse manufacturing domains.

2. Conversational AI Model Development Module:

- **Language Model Integration**: Integrate state-of-the-art language models, such as generative pretrained models like GPT-2, within the conversational AI framework. Customize these models to align with the nuances of human-robot collaboration in manufacturing.
- **Intent Recognition Module**: Develop a module focused on accurately recognizing user intentions from dialogue segments. This involves training models to understand and interpret the objectives or requests expressed by users in manufacturing contexts.
- **Response Generation Module**: Design a module to generate contextually appropriate and task-relevant responses based on recognized user intentions. This includes ensuring that the responses align with the objectives of human-robot collaboration in manufacturing tasks.
- **3. Wizard-of-Oz Technique Implementation Module:**
- **Noise Reduction Techniques:** Implement techniques within the Wizard-of-Oz module to reduce noise in the dataset. This could involve refining the human-controlled simulation process to create more authentic and representative dialogues.

4. Benchmarking and Evaluation Module:

- **Performance Metrics Definition**: Develop a module to define and measure performance metrics for evaluating the effectiveness of the IRWoZ dataset and the conversational AI models. Metrics may include accuracy in dialogue state tracking and response coherence.

- **Comparison with Baseline Models:** Compare the performance of conversational AI models trained on the IRWoZ dataset against baseline models, highlighting improvements or specific advantages gained from the dataset's industrial focus.

5. User Interface (UI) Integration Module:

- **Interaction**: Design a user interface module that facilitates interaction between users and the conversational AI system. This module can be tailored to the unique needs of human-robot collaboration in manufacturing, ensuring a seamless and intuitive user experience.

3.3 TECHNIQUE USED OR ALGORITHM USED

3.3.1 EXISTING TECHNIQUE: -

A novel deep neural network model, TT-Net, is proposed for conversational reading comprehension.

It is able to capture topic transfer features in the conversation history by temporal convolution network.

Although it is the first attempt to leverage TCN structure to solve conversation reading comprehension tasks, our proposed model shows pretty good performance in the CoQA dataset and performs better than several baseline models including the strong model FlowQA.

DRAWBACKS:

Interpretability Challenges:

Deep neural network models, especially those incorporating complex structures like temporal convolution networks (TCN), may lack interpretability. Understanding how the model arrives at its answers or the specific features it uses for decision-making could be challenging, impacting the model's transparency.

Data Dependency:

Deep neural networks, including TT-Net, often require large amounts of labeled data for effective training. The model's performance may be highly dependent on the availability and quality of the training data, and its generalizability to diverse conversational contexts may be limited by the dataset it was trained on.

Topic Transfer Limitations:

While TT-Net is designed to capture topic transfer features in conversation history, its effectiveness might be constrained by the complexity and variability of language. It may struggle with subtle shifts in context or topics that deviate significantly from the training data, affecting its adaptability to diverse conversational scenarios.

Computational Intensity:

Deep neural networks can be computationally intensive, and models like TT-Net, which leverage temporal convolution networks, may require significant computational resources during training and inference. This could limit its practical applicability in resource-constrained environments.

Overfitting Concerns:

Deep neural networks are susceptible to overfitting, especially when dealing with limited datasets. The TT-Net model may memorize specific patterns in the training data that do not generalize well to new, unseen conversational contexts, impacting its performance on out-of-sample datasets.

3.3.2 PROPOSED TECHNIQUE USED OR ALGORITHM USED:

- Feed forward neural network for multi-class classification using the Keras library with a Tensor Flow backend.
- The first layer has 128 neurons with a ReLU activation function.
- The second layer has 64 neurons with a ReLU activation function.
- The third layer (output layer) contains a number of neurons equal to the number of intents to predict the output intent using the softmax activation function. The softmax function is commonly used for multi-class classification problems as it converts raw scores into probabilities, making it easier to interpret the results.

ADVANTAGES:

1. Ease of Implementation with Keras:

Leveraging the Keras library simplifies the implementation process, providing a high-level neural network API that allows for rapid prototyping. Keras's user-friendly interface facilitates the construction of complex neural network architectures with relatively concise code.

2. TensorFlow Backend Efficiency:

Utilizing TensorFlow as the backend ensures efficiency and scalability. TensorFlow is a widely adopted and well-supported deep learning framework, offering optimized computation and parallelization, which can significantly enhance training speed and model performance.

3. ReLU Activation Functions for Non-linearity:

The use of Rectified Linear Unit (ReLU) activation functions in the first and second layers introduces non-linearity to the model. ReLU is known for addressing the vanishing gradient problem and accelerating convergence during training.

4. Model Depth and Neuronal Hierarchy:

The model incorporates multiple layers, enhancing its ability to learn hierarchical representations and abstract features from the input data. The 128 neurons in the first layer and 64 neurons in the second layer allow the network to capture complex patterns and relationships within the data.

5. Softmax Activation for Multi-class Classification:

The choice of the softmax activation function in the output layer is advantageous for multi-class classification tasks. Softmax converts raw scores into probability distributions, facilitating the interpretation of the model's predictions. It ensures that the sum of output probabilities across all classes is equal to one, making it easier to understand the certainty of predictions.

.

CHAPTER 4

REQUIREMENTS ENGINEERING

4.1 GENERAL

We can see from the results that on each database, the error rates are very low due to the discriminatory power of features and the regression capabilities of classifiers. Comparing the highest accuracies (corresponding to the lowest error rates) to those of previous works, our results are very competitive.

4.2 HARDWARE REQUIREMENTS

The hardware requirements may serve as the basis for a contract for the implementation of the system and should therefore be a complete and consistent specification of the whole system. They are used by software engineers as the starting point for the system design. It should what the system do and not how it should be implemented.

- PROCESSOR : INTEL CORE i3
- RAM : 4GB DD RAM
- HARD DISK : 250 GB

4.3 SOFTWARE REQUIREMENTS

The software requirements document is the specification of the system. It should include both a definition and a specification of requirements. It is a set of what the system should do rather than how it should do it. The software requirements provide a basis for creating the software requirements specification. It is useful in estimating cost, planning team activities, performing tasks and tracking the teams and tracking the team's progress throughout the development activity.

- Operating System : Windows 7/8/10
- Platform : Spyder3
- Programming Language : Python
- Front End : Spyder3

4.4 FUNCTIONAL REQUIREMENTS

A functional requirement defines a function of a software-system or its component. A function is described as a set of inputs, the behavior, Firstly, the system is the first that achieves the standard notion of semantic security for data confidentiality in attribute-based deduplication systems by resorting to the hybrid cloud architecture.

4.5 NON-FUNCTIONAL REQUIREMENTS

The major non-functional Requirements of the system are as follows

Usability

The system is designed with completely automated process hence there is no or less user intervention.

Reliability

The system is more reliable because of the qualities that are inherited from the chosen platform python. The code built by using python is more reliable.

Performance

This system is developing in the high level languages and using the advanced back-end technologies it will give response to the end user on client system with in very less time.

Supportability

The system is designed to be the cross platform supportable. The system is supported on a wide range of hardware and any software platform, which is built into the system.

Implementation

The system is implemented in web environment using Jupyter notebook software. The server is used as the intellignce server and windows 10 professional is used as the platform. Interface the user interface is based on Jupyter notebook provides server system.

CHAPTER 5

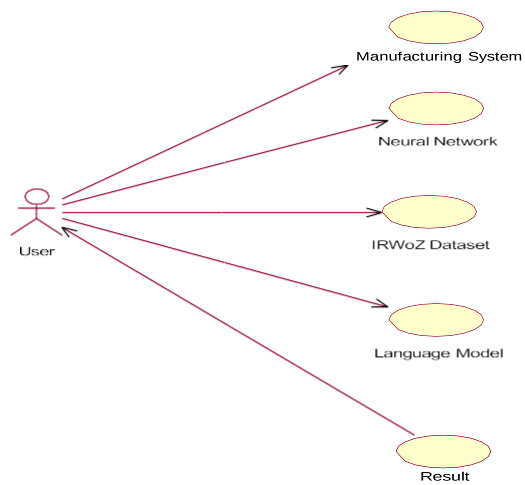
DESIGN ENGINEERING

5.1 GENERAL

Design Engineering deals with the various UML [Unified Modelling language] diagrams for the implementation of project. Design is a meaningful engineering representation of a thing that is to be built. Software design is a process through which the requirements are translated into representation of the software. Design is the place where quality is rendered in software engineering.

5.2 UML DIAGRAMS

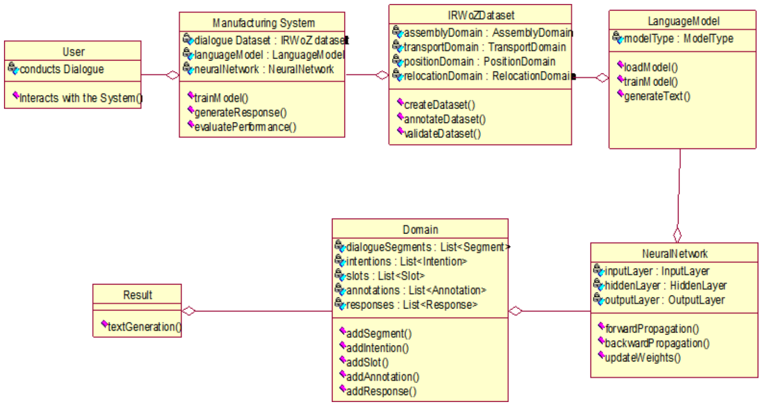
5.2.1 USE CASE DIAGRAM



EXPLANATION:

The main purpose of a use case diagram is to show what system functions are performed for which actor. Roles of the actors in the system can be depicted. The above diagram consists of user as actor. Each will play a certain role to achieve the concept.

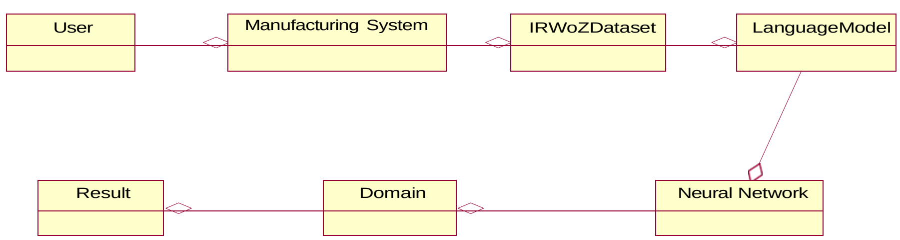
5.2.2 CLASS DIAGRAM



EXPLANATION

In this class diagram represents how the classes with attributes and methods are linked together to perform the verification with security. From the above diagram shown the various classes involved in our project.

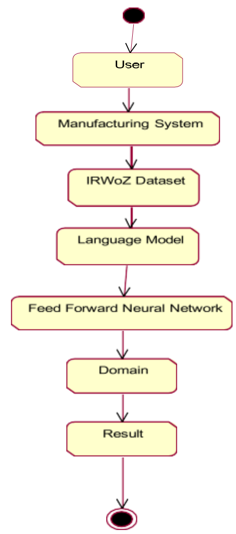
5.2.3 OBJECT DIAGRAM



EXPLANATION:

In the above digram tells about the flow of objects between the classes. It is a diagram that shows a complete or partial view of the structure of a modeled system. In this object diagram represents how the classes with attributes and methods are linked together to perform the verification with security.

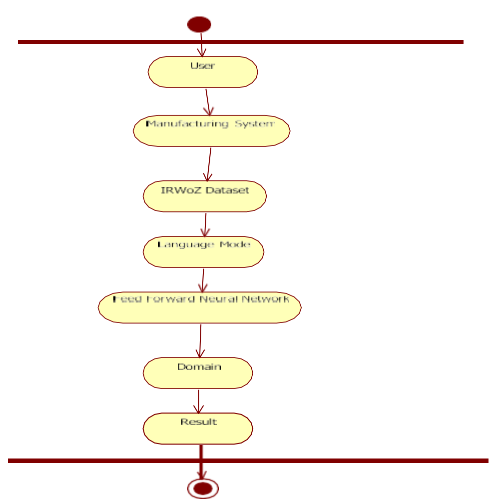
5.2.4 STATE DIAGRAM



EXPLANATION:

State diagram are a loosely defined diagram to show workflows of stepwise activities and actions, with support for choice, iteration and concurrency. State diagrams require that the system described is composed of a finite number of states; sometimes, this is indeed the case, while at other times this is a reasonable abstraction. Many forms of state diagrams exist, which differ slightly and have different semantics.

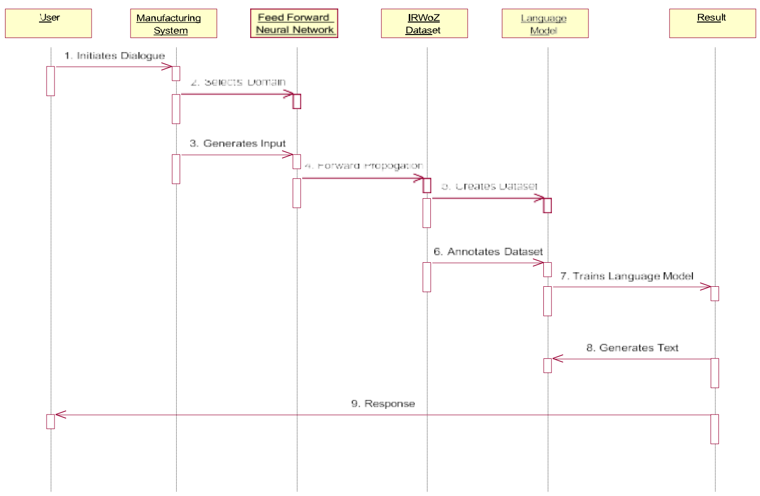
5.2.5 ACTIVITY DIAGRAM:



EXPLANATION:

Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. In the Unified Modeling Language, activity diagrams can be used to describe the business and operational step-by-step workflows of components in a system. An activity diagram shows the overall flow of control.

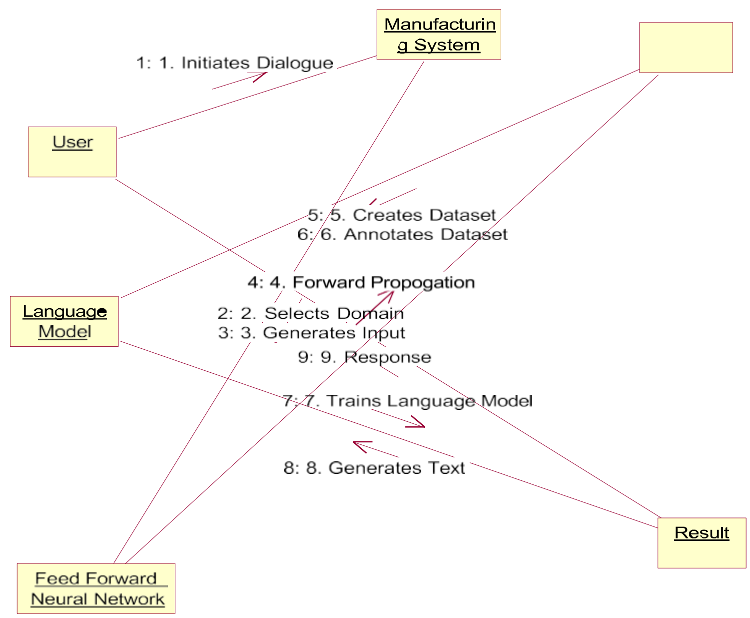
5.2.6 SEQUENCE DIAGRAM



EXPLANATION:

A sequence diagram in Unified Modeling Language (UML) is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. A sequence diagram shows object interactions arranged in time sequence. It depicts the objects and classes involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario.

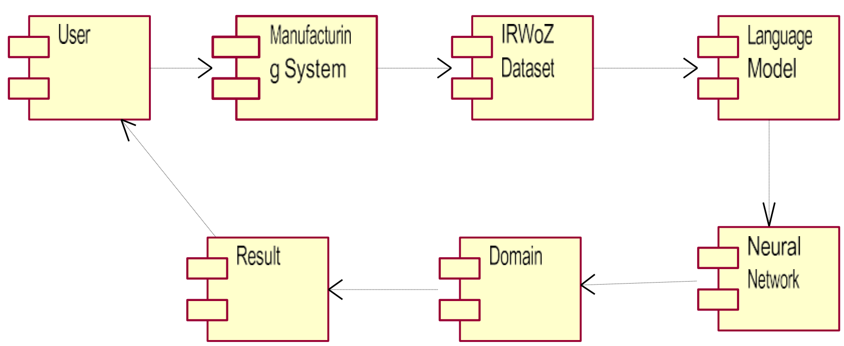
5.2.7 COLLABORATION DIAGRAM



EXPLANATION:

A collaboration diagram, also called a communication diagram or interaction diagram, is an illustration of the relationships and interactions among software objects in the Unified Modeling Language (UML). The concept is more than a decade old although it has been refined as modeling paradigms have evolved.

5.2.8 COMPONENT DIAGRAM

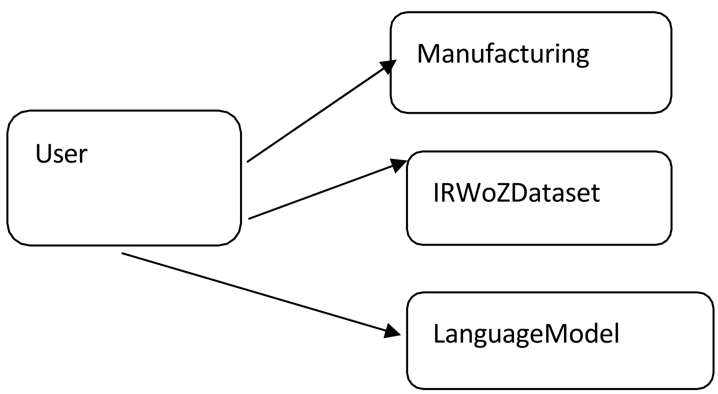


EXPLANATION

In the Unified Modeling Language, a component diagram depicts how components are wired together to form larger components and or software systems. They are used to illustrate the structure of arbitrarily complex systems. User gives main query and it converted into sub queries and sends through data dissemination to data aggregators. Results are to be showed to user by data aggregators. All boxes are components and arrow indicates dependencies.

5.2.9 DATA FLOW DIAGRAM

Level 0



Level 1

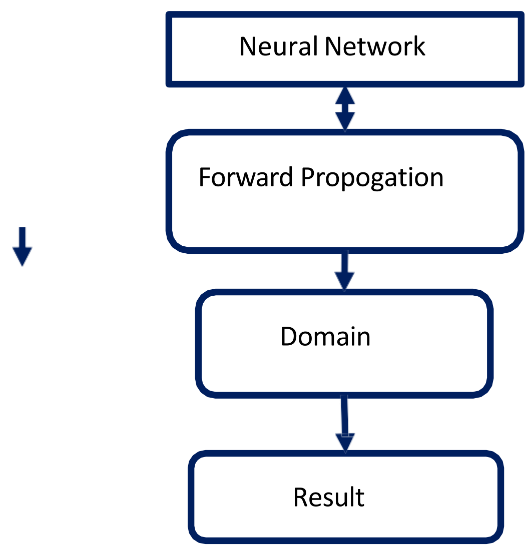
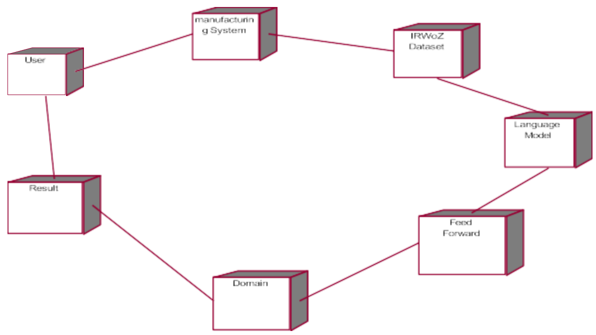


Fig 4.9: Data Flow Diagrams

EXPLANATION:

A data flow diagram (DFD) is a graphical representation of the "flow" of data through an information system, modeling its process aspects. Often they are a preliminary step used to create an overview of the system which can later be elaborated. DFDs can also be used for the visualization of data processing (structured design).

5.2.10 DEPLOYMENT DIAGRAM



EXPLANATION:

Deployment Diagram is a type of diagram that specifies the physical hardware on which the software system will execute. It also determines how the software is deployed on the underlying hardware. It maps software pieces of a system to the device that are going to execute it.

5.2.11 SYSTEM ARCHITECTURE:

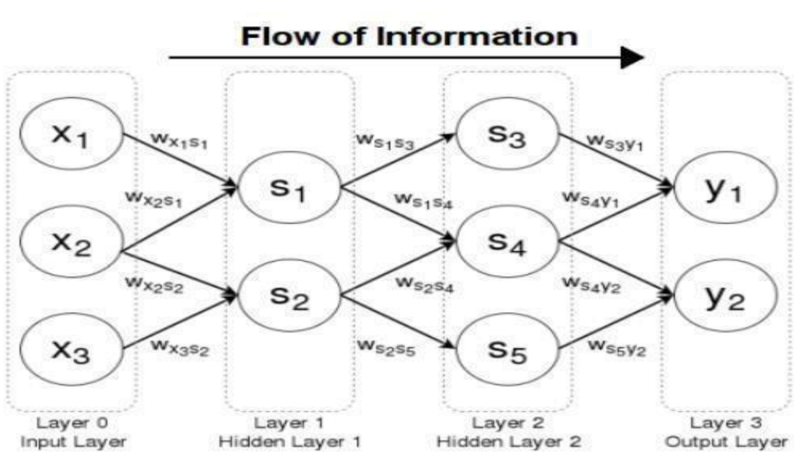


Fig 4.11: Feed forward neural network

CHAPTER 6

DEVELOPMENT TOOLS

6.1 Python

Python is a high-level, interpreted, interactive and object-oriented scripting language. Python is designed to be highly readable. It uses English keywords frequently where as other languages use punctuation, and it has fewer syntactical constructions than other languages.

6.2 HistoryofPython

Python was developed by Guido van Rossum in the late eighties and early nineties at the National Research Institute for Mathematics and Computer Science in the Netherlands.

Python is derived from many other languages, including ABC, Modula-3, C, C++, Algol-68, SmallTalk, and Unix shell and other scripting languages.

Python is copyrighted. Like Perl, Python source code is now available under the GNU General Public License (GPL).

Python is now maintained by a core development team at the institute, although Guido van Rossum still holds a vital role in directing its progress.

6.3 Importance of Python

- **Python is Interpreted** − Python is processed at runtime by the interpreter. You do not need to compile your program before executing it. This is similar to PERL and PHP.
- **Python is Interactive** − You can actually sit at a Python prompt and interact with the interpreter directly to write your programs.
- **Python is Object-Oriented** − Python supports Object-Oriented style or technique of programming that encapsulates code within objects.

- **Python is a Beginner's Language** − Python is a great language for the beginner-level programmers and supports the development of a wide range of applications from simple text processing to WWW browsers to games.

6.4 Features of Python

- **Easy-to-learn** − Python has few keywords, simple structure, and a clearly defined syntax. This allows the student to pick up the language quickly.
- **Easy-to-read** − Python code is more clearly defined and visible to the eyes.
- **Easy-to-maintain** − Python's source code is fairly easy-to-maintain.
- **A broad standard library** − Python's bulk of the library is very portable and cross-platform compatible on UNIX, Windows, and Macintosh.
- **Interactive Mode** − Python has support for an interactive mode which allows interactive testing and debugging of snippets of code.
- **Portable** − Python can run on a wide variety of hardware platforms and has the same interface on all platforms.
- **Extendable** − You can add low-level modules to the Python interpreter. These modules enable programmers to add to or customize their tools to be more efficient.
- **Databases** − Python provides interfaces to all major commercial databases.
- **GUI Programming** − Python supports GUI applications that can be created and ported to many system calls, libraries and windows systems, such as Windows MFC, Macintosh, and the X Window system of Unix.
- **Scalable** − Python provides a better structure and support for large programs than shell scripting.

Apart from the above-mentioned features, Python has a big list of good features, few are listed below −

- It supports functional and structured programming methods as well as OOP.
- It can be used as a scripting language or can be compiled to byte-code for building large applications.
- It provides very high-level dynamic data types and supports dynamic type checking.
- IT supports automatic garbage collection.
- It can be easily integrated with C, C++, COM, ActiveX, CORBA, and Java.

6.5 Libraries used in python

- numpy - mainly useful for its N-dimensional array objects.
- pandas - Python data analysis library, including structures such as dataframes.
- matplotlib - 2D plotting library producing publication quality figures.
- scikit-learn - the machine learning algorithms used for data analysis and data mining tasks.

Figure : NumPy, Pandas, Matplotlib, Scikit-learn

CHAPTER 7

IMPLEMENTATION & SNAPSHOTS

7.1 GENERAL

Coding:

```
{
    "intents": [{
            "tag": "greetings",
            "patterns": ["hi there", "hello","haroo","yaw","wassup", "hi", "hey", "holla", "hello"],
            "responses": ["hello thanks for checking in", "hi there, how can i help you"],
            "context": [""]
        },
        {
            "tag": "goodbye",
            "patterns": ["bye", "good bye", "see you later"],
            "responses": ["have a nice time, welcome back again", "bye bye"],
            "context": [""]
        },
        {
            "tag": "thanks",
            "patterns": ["Thanks", "okay","Thank you","thankyou", "That's helpful", "Awesome, thanks",
"Thanks for helping me", "wow", "great"],
            "responses": ["Happy to help!", "Any time!","you're welcome", "My pleasure"],
            "context": [""]
        },
        {
```

"tag": "noanswer",

"patterns": [""],

"responses": ["Sorry, I didn't understand you", "Please give me more info", "Not sure I understand that"],

"context": [""]

},

{

"tag": "need",

"patterns": ["I need you", "All I need is you","I want you"],

"responses": ["Yes I'm here to assist you"],

"context": [""]

},

{

"tag": "stupid",

"patterns": [" Robots are stupid", " Are you stupid"],

"responses": [" No, we are superintelligent.", " No, lots of people improve my brain."],

"context": [""]

},

{

"tag": "lie",

"patterns": [" Robots are not allowed to lie"],

"responses": [" Sure we are. We choose not to.", " Only if we're programmed to.", " A robot has its own free will, you know."],

"context": [""]

},

{

"patterns": [" What kind of hardware"],

```
    "responses": [" I work on all kinds of computers, Mac, IBM or UNIX. it doesn't matter to me."],
    "context": [""]
  },
  {
    "tag": "bot1",
    "patterns": ["are you a bot"],
    "responses": ["Yes. I work and all my operations are based on the internet servers."],
    "context": [""]
  },
  {
    "tag": "best_sellers",
    "patterns": ["Can you recommend some best-selling products?", "Best sellers"],
    "responses": ["Certainly! Our best sellers include [Product A], [Product B], and [Product C]."],
    "context": [""]
  },
  {
    "tag": "shipping_timeframe",
    "patterns": ["How long does shipping usually take?", "Shipping timeframe"],
    "responses": ["Our standard shipping takes 3-5 business days, while express shipping delivers within 1-2 business days."],
    "context": [""]
  },
  {
    "tag": "shipping_cost",
    "patterns": ["What are the shipping costs?", "Shipping cost"],
    "responses": ["Shipping costs vary depending on the shipping method and destination. Please proceed to checkout to view the exact shipping cost for your order."],
```

"context": [""]

},

{

"tag": "product_categories",

"patterns": ["What types of products do you offer?", "Product categories"],

"responses": ["We offer a wide range of products including electronics, clothing, accessories, home goods, and more."],

"context": [""]

},

{

"tag": "best_sellers",

"patterns": ["Can you recommend some best-selling products?", "Best sellers"],

"responses": ["Certainly! Our best sellers include [Product A], [Product B], and [Product C]."],

"context": [""]

},

{

"tag": "product_recommendations_7",

"patterns": ["Need help choosing products?", "Product selection help"],

"responses": ["Let us assist you with our recommended products: [Recommended Product A], [Recommended Product B], [Recommended Product C]."],

"context": [""]

},

{

"tag": "shipping_updates",

"patterns": ["How can I track my shipment?", "Shipping tracking"],

"responses": ["You can track your shipment using the tracking number provided in your order confirmation email."],

"context": [""]

},

{

"tag": "product_returns",

"patterns": ["What is your return policy?", "Return policy"],

"responses": ["Our return policy allows returns within 30 days of purchase. Please refer to our returns page for more details."],

"context": [""]

},

{

"tag": "delivery_timeframe",

"patterns": ["How long does delivery take?", "Delivery timeframe"],

"responses": ["Delivery times vary depending on the shipping method and destination. Standard delivery typically takes 3-5 business days, while express delivery can be as fast as 1-2 business days."],

"context": [""]

},

{

"tag": "delivery_confirmation",

"patterns": ["How will I know when my delivery is confirmed?", "Delivery confirmation"],

"responses": ["You will receive a confirmation email or SMS once your delivery has been successfully completed. You can also track the status of your delivery using the provided tracking information."],

"context": [""]

},

}

]}

General:

This project is implements like application using python and the Server process is maintained using the SOCKET & SERVERSOCKET and the Design part is played by Cascading Style Sheet.

SNAPSHOTS

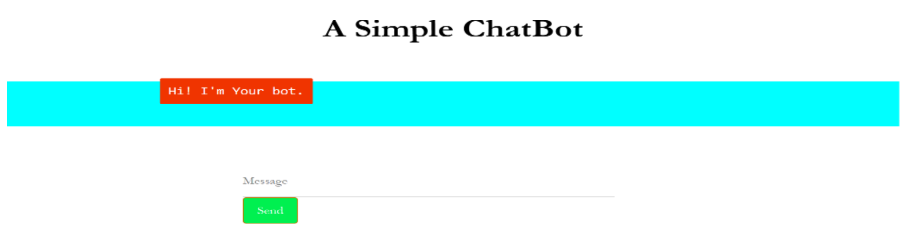


Fig1. A Simple ChatBot

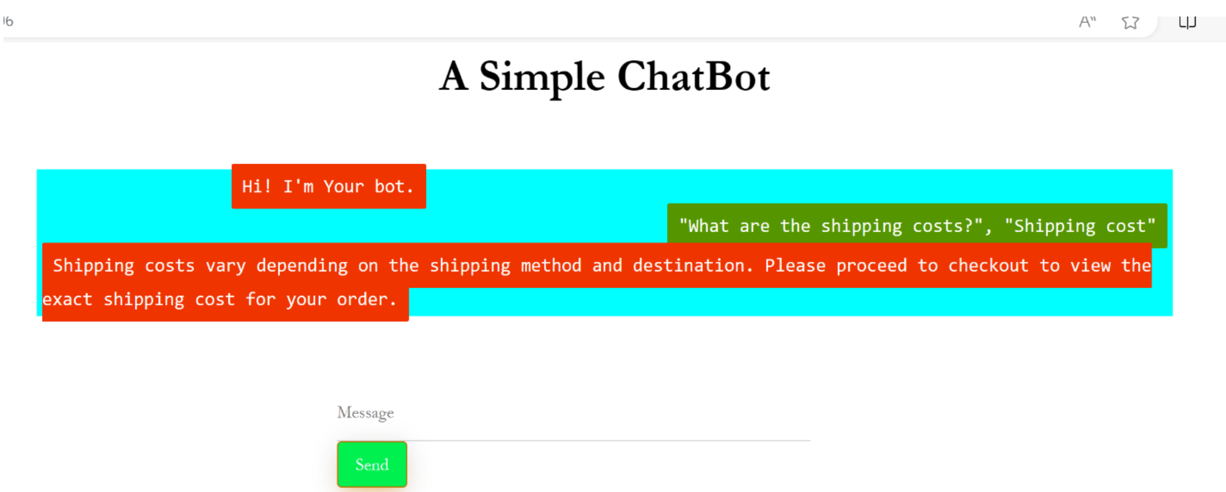


Fig2.A Simple ChatBot

CHAPTER 8
SOFTWARE TESTING

8.1 GENERAL

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub assemblies, assemblies and/or a finished product It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

8.2 DEVELOPING METHODOLOGIES

The test process is initiated by developing a comprehensive plan to test the general functionality and special features on a variety of platform combinations. Strict quality control procedures are used. The process verifies that the application meets the requirements specified in the system requirements document and is bug free. The following are the considerations used to develop the framework from developing the testing methodologies.

8.3 Types of Tests

8.3.1 Unit testing

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program input produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .it is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

8.3.2 Functional test

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is centered on the following items:

Valid Input : identified classes of valid input must be accepted.

Invalid Input : identified classes of invalid input must be rejected.

Functions : identified functions must be exercised.

Output : identified classes of application outputs must be exercised.

Systems/Procedures: interfacing systems or procedures must be invoked.

8.3.3 System Test

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

8.3.4 Performance Test

The Performance test ensures that the output be produced within the time limits,and the time taken by the system for compiling, giving response to the users and request being send to the system for to retrieve the results.

8.3.5 Integration Testing

Software integration testing is the incremental integration testing of two or more integrated software components on a single platform to produce failures caused by interface defects.

The task of the integration test is to check that components or software applications, e.g. components in a software system or – one step up – software applications at the company level – interact without error.

8.3.6 Acceptance Testing

User Acceptance Testing is a critical phase of any project and requires significant participation by the end user. It also ensures that the system meets the functional requirements.

Acceptance testing for Data Synchronization:

- The Acknowledgements will be received by the Sender Node after the Packets are received by the Destination Node
- The Route add operation is done only when there is a Route request in need
- The Status of Nodes information is done automatically in the Cache Updation process

8.2.7 Build the test plan

Any project can be divided into units that can be further performed for detailed processing. Then a testing strategy for each of this unit is carried out. Unit testing helps to identity the possible bugs in the individual component, so the component that has bugs can be identified and can be rectified from errors.

CHAPTER 9

CONCLUSION

CONCLUSION

In conclusion, the development of a system integrating conversational artificial intelligence (AI) and a feed forward neural network for human-robot collaboration in manufacturing environments represents a significant stride toward fostering adaptive and seamless interactions. The amalgamation of advanced language models, dialogue datasets tailored for industrial contexts, and neural network capabilities opens new avenues for enhancing flexibility and user experience in manufacturing tasks.

The introduction of the Industrial Robot Wizard-of-Oz Dialoguing Dataset (IRWoZ) addresses a critical gap by providing a specialized resource meticulously crafted to empower human-robot collaboration (HRC). Encompassing distinct domains such as assembly, transportation, position, and relocation, the dataset serves as a valuable tool for training effective conversational AI models. The utilization of the Wizard-of-Oz technique ensures authenticity and mitigates noise, laying the foundation for more accurate and context-aware dialogues. Looking ahead, future enhancements could encompass advancements in natural language processing, dynamic domain expansion, real-time interaction capabilities, multi-modal interactions, and the integration of user feedback mechanisms. These developments aim to not only improve the system's technical capabilities but also to ensure its adaptability to evolving manufacturing needs and industry standards.

As the manufacturing landscape continues to evolve, the seamless integration of conversational AI and neural networks holds the promise of not only optimizing production processes but also fostering a collaborative and user-friendly environment on the shop floor. The pursuit of innovation in this domain underscores a commitment to pushing the boundaries of human-robot collaboration and contributing to the ongoing transformation of industrial practices.

CHAPTER 10

FUTURE ENHANCEMENT

10.1 FUTURE ENHANCEMENTS:

Future enhancements for a system involving conversational AI, human-robot collaboration in manufacturing, and a feed forward neural network could include advancements in various aspects of the system's functionality, scalability, and adaptability. Here are some potential future enhancements:

Advanced Natural Language Processing (NLP): Enhance the capabilities of the language model by incorporating more sophisticated natural language processing techniques. This may involve adopting state-of-the-art pre-trained language models, exploring transformer-based architectures, or integrating domain-specific language models for improved dialogue understanding and generation.

Dynamic Domain Expansion: Enable the system to dynamically adapt to new manufacturing domains without extensive manual intervention. This could involve developing mechanisms for on-the-fly dataset creation, annotation, and model retraining to seamlessly integrate new tasks or manufacturing processes.

Real-time Interaction: Improve the system's responsiveness for real-time human-robot collaboration. Consider optimizations and parallel processing techniques to reduce latency in generating responses and processing dialogue inputs.

Multi-modal Interaction: Explore multi-modal capabilities by incorporating additional sensory inputs, such as visual or tactile information, to enhance the system's understanding of the manufacturing environment. This could involve integrating computer vision or other sensor technologies.

User Feedback Integration: Implement mechanisms to collect and utilize user feedback for continuous improvement. Incorporate user feedback loops to enhance dialogue understanding, refine response generation, and adapt the system to the evolving needs of shop floor worker

REFERENCES

[1]O. Madsen, C. Bro Sørensen, R. Larsen, L. Overgaard, and N. J. Jacobsen, "A system for complex robotic welding," Ind. Robot, Int. J., vol. 29, no. 2, pp. 127–131, Apr. 2002.

[2]A. S. Olesen, B. B. Gergaly, E. A. Ryberg, M. R. Thomsen, and D. Chrysostomou, "A collaborative robot cell for random bin-picking based on deep learning policies and a multi-gripper switching strategy," Proc. Manuf., vol. 51, pp. 3–10, Jan. 2020.

[3]J. F. Buhl, R. Grønhøj, J. K. Jørgensen, G. Mateus, D. Pinto, J. K. Sørensen, S. Bøgh, and D. Chrysostomou, "A dual-arm collaborative robot system for the smart factories of the future," Proc. Manuf., vol. 38, pp. 333–340, Jan. 2019.

[4]E. R. da Silva, C. Schou, S. Hjorth, F. Tryggvason, and M. S. Sørensen, "Plug & produce robot assistants as shared resources: A simulation approach," J. Manuf. Syst., vol. 63, pp. 107–117, Apr. 2022.

[5]S. Bogh, "Integration and assessment of multiple mobile manipulators in a real-world industrial production facility," in Proc. ISR/Robotik 41st Int. Symp. Robot., Jun. 2014, pp. 1–8.

[6]S. Hjorth, J. Lachner, S. Stramigioli, O. Madsen, and D. Chrysostomou, "An energy-based approach for the integration of collaborative redundant robots in restricted work environments," in Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS), Oct. 2020, pp. 7152–7158.

[7]C. Li and H. J. Yang, "Bot-X: An AI-based virtual assistant for intelligent manufacturing," Multiagent Grid Syst., vol. 17, no. 1, pp. 1–14, Apr. 2021.

[8]M. Dibitonto, K. Leszczynska, F. Tazzi, and C. M. Medaglia, "Chatbot in a campus environment: Design of LiSA, a virtual assistant to help students in their university life," in Proc. Int. Conf. Hum.-Comput. Interact., Cham, Switzerland: Springer, 2018, pp. 103–116.