LIST OF ABBREVIATIONS

NLP: Natural language processing

NLG:Natural language generation

NLU: Natural language understanding

**Part.1:Overview of the problem of using chatbots in human-machine interaction**

There are several challenges and potential problems when using chatbots in human-machine interaction.

**Limited understanding**: Chatbots are limited in their understanding of language and context, so they may not be able to accurately interpret or respond to certain types of input. This can lead to misunderstandings and frustration for users.

**Lack of empathy**: Chatbots are not capable of experiencing emotions, so they may not be able to effectively communicate empathy or respond appropriately to emotional cues. This can make them feel less human and less able to provide a satisfying customer experience.

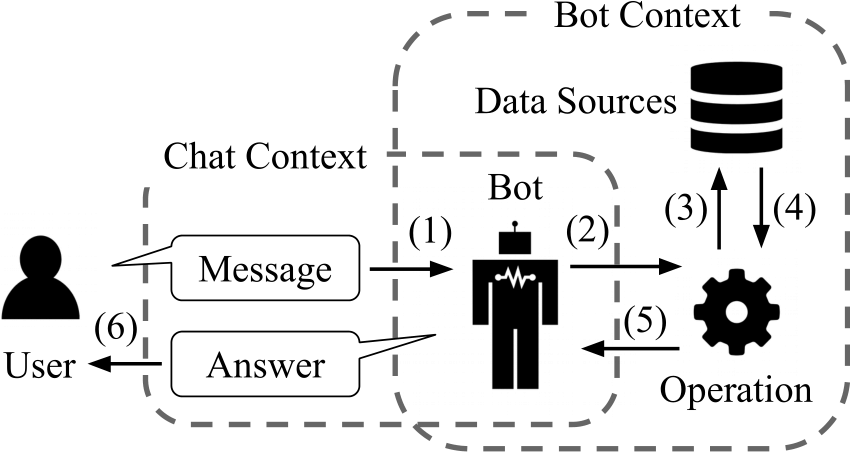
**Inflexibility**: Chatbots are often designed to follow a specific set of rules or scripts, which can make them inflexible and unable to handle unexpected input or situations. This can make them feel rigid and difficult to use.

**Privacy concerns**: Chatbots may collect and store user data, which can raise privacy concerns if the data is not properly secured or if it is used for purposes other than those for which it was intended.

**Misuse**: Chatbots may be used for malicious purposes, such as spamming or phishing, or to spread misinformation or propaganda.

Despite these challenges, chatbots can still be useful tools for human-machine interaction, particularly when they are carefully designed and used in appropriate contexts.

**Figure:1.3**



**Chatbot don’t take actions**

One of the struggles of simple chatbot technology is the inability to complete backend processes. Instead, they rely on the end-user to take further actions themselves.

Part of the reason for that is in many cases, contact centre agents use between 5 and 8 different systems to resolve inbound queries.

The result is that the customer is frustrated because they either have to specifically request a human agent or give up.

1.1: **Chatbots. Basic concepts and definitions**

“What is this Chatbot?”

Chatbots are chatbots that users communicate with as if they are writing to a human through messaging in the digital environment and that they use for various purposes such as getting information about a subject and taking action. Using a chatbot, it is possible to order pizza, buy movie tickets, check-in for flight, find out account balance.

**There are 2 Types of Chatbots**

1-**Basic Chatbot**:

In simple chats, the chatbot works based on pre-written keywords. Each of these given commands must be written separately by the developer using regular expressions or other forms of string analysis.

If the user has asked a question without using a single keyword, the robot cannot understand it and, as a rule, responds with messages such as "sorry, I don't understand".

2-**Artificial Intelligence Powered Chatbot**:

Intelligent chatbot relies on artificial intelligence when communicating with users. Instead of pre-prepared answers, the robot gives satisfactory answers on the topic. Also, all words spoken by users are saved for further processing.

**Figure 1.1**

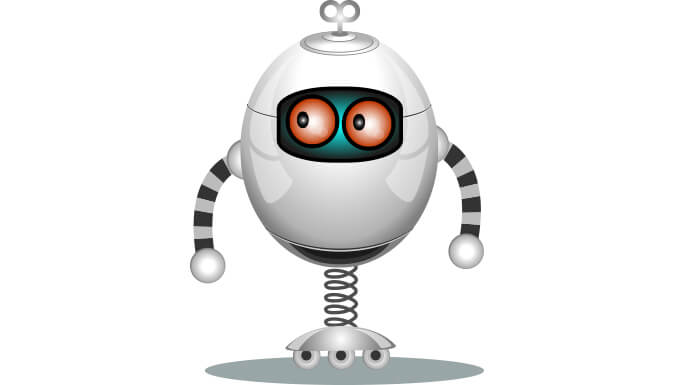


Intelligent chatbot relies on artificial intelligence when communicating with users. Instead of pre-prepared answers, the robot gives satisfactory answers on the topic. Also, all words spoken by users are saved for further processing.

However, the report published by the US-based Forrester institution indicates that artificial intelligence is not magic and is not yet ready to produce amazing experiences for users. On the contrary, a lot of work is required:

“Chat developers and designers need to guide conversations with soft fixes. These designers also need to encourage or even force clients to keep up with the sub-core experiences throughout the process. Because learning is really slow.”

**Figure:1.2**



There are several search engines for bots such as Chatbottle, Botlist, and Thereisabotforthat: for example, it helps developers notify users about the launch of new conversation bots.

**1.2 Using chatbots in the context of human resource**

The use of chatbot for HR is very important and in this context, chatbot development directly affects HR. As I explained in the previous sections, it can answer any simple question (if it is registered in the chatbot's database), but the use of artificial intelligence and chatbot directly affects HR and maybe in the future HR staff can be replaced by chatbots

**1.3 Overview of the process of human resource support requests set in natural language**

In the world where competition is increasing rapidly with the development of technology, HR managers; collection of accurate data on personnel and work process, personnel training and recruitment, orientation process, and performance appraisal etc. They use artificial intelligence technologies to facilitate difficult functions. Artificial intelligence enables HR Managers to do their jobs faster and more efficiently. Artificial intelligence is also used in various departments such as human resources department, finance department, marketing and production department. This study deals with human resource management in a historical perspective. The main purpose of the study is to examine the role of artificial intelligence in human resource management. However, it tries to address how AI supports Human Resources functions such as recruitment, screening and interview process, training and talent management. As a result of the study, ideas about the intersection of artificial intelligence and human resource management cases, and some institutional and academic suggestions for the future are included.

1.4 Overview of existing software for query analysis in natural language  
  
**NLP Technology Overview**

Machine learning models for NLP: We mentioned earlier that modern NLP is heavily based on an approach to artificial intelligence called machine learning. Machine learning makes predictions by generalizing samples from a dataset. This dataset is called training data, and machine learning algorithms are trained on this training data to create a machine learning model that performs a target task.

For example, sentiment analysis training data consists of sentences containing emotion (for example, positive, negative, or neutral emotion). A machine learning algorithm reads this dataset and generates a model that takes sentences as input and returns emotions. Such models, which take sentences or documents as input and introduce a label for that input, are called document classification models. Document classifiers can also be used to classify documents based on the topics they cover (for example, as sports, finance, politics, etc.).

Another model is used to recognize and classify entities in documents. For each word in the document, the model predicts whether the word is an entity expression and, if so, what type of entity is being referred to. For example, in the sentence "Share of Company XYZ traded for $28 yesterday," "Company XYZ" is a company asset, "$28" is a currency amount, and "yesterday" is a date. Training data for entity recognition is a collection of text where each word is labeled as it relates to whatever type of entity. Such patterns that generate a tag for each word in the input are called string tagging patterns.

Series-to-series models are one of the newest members of the model family used in NLP. The string-to-array (or seq2seq) model takes an entire sentence or document as input (just like a document classifier) but produces a sentence or other sequence (for example, a computer program) as output. (The document classifier, on the other hand, produces only a single symbol as output.) Examples of Seq2seq models include machine translation, document summarization (the output is a summary of the input), and semantic parsing (the input is a summary of the input) that takes an English sentence as input and returns a French sentence. is a query or request, and the output is a computer program that implements the request).

**Figure:NLP**



Deep learning, pre-trained models and transfer learning: Deep learning is the most widely used type of machine learning in NLP. In the 1980s, researchers developed neural networks. In these neural networks, multiple primitive machine learning models converge into a single network. Similar to the brain, simple machine learning models are sometimes referred to as "neurons." These neurons are arranged in layers, and the deep neural network contains many layers. Deep learning is machine learning that uses deep neural network models.

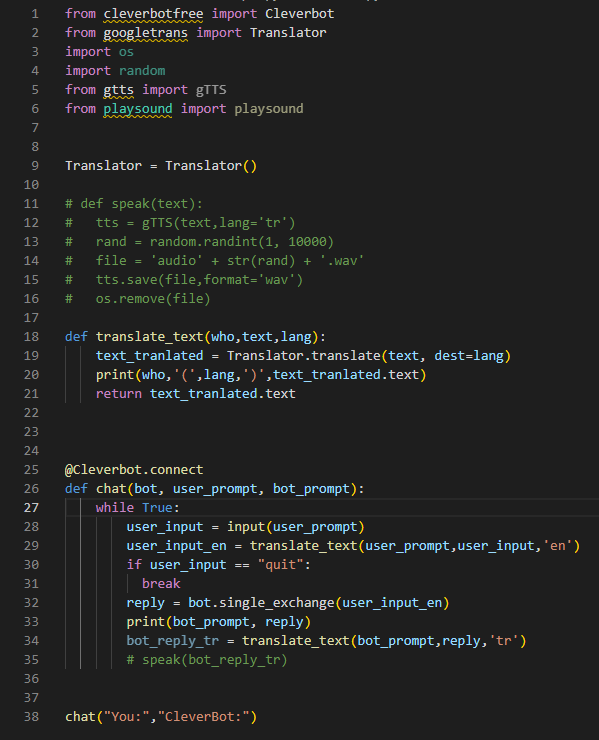
Because of their complexity, training a deep neural network requires a lot of data, and processing this data requires a lot of computing power and time. Modern deep neural network NLP models are trained from a variety of sources such as Wikipedia and data from the web. Training data can be 10 GB or larger and may require a week or longer to train the deep neural network on a high-performance cluster. (Researchers have found that training even deeper models from even larger datasets performs better. So there is a race to train even larger models from larger datasets.)

The extensive data and computing requirements of Deep Neural Networks may seem to severely limit their usefulness. However, transfer learning enables a trained deep neural network to be more trained to perform a new task with much less training data and computing effort. The simplest type of transfer learning is called fine tuning. It basically means first training the model on a large general dataset (e.g. Wikipedia) and then training ("tweaking") the model on a much smaller task-specific dataset with the actual target task tag. Surprisingly, fine-tuning datasets can be extremely small, consisting of only hundreds or even dozens of training samples, and fine-tuning training can be performed in minutes on a single CPU. Transfer learning facilitates company-wide deployment of deep learning models.

There is now an extensive ecosystem of providers offering pre-trained deep learning models trained in different combinations of languages, datasets, and pretraining tasks. These pre-trained models can be downloaded and fine-tuned for a wide range of target tasks.

**Part 2. Selecting tools and creating a chatbot for human resource**

I was wrote code for example

(Figure 2.1: Code)  
  
(That’s example not a Project)

I use cleverbot and Google translate pack   
Let’s explain what’s this:  
Cleverbot: Cleverbot.com used to have a free API for their chatbot application. They have

removed their free API in place of a tiered subscription API service.

cleverbotfree is a free alternative to that API that uses a headless Firefox

browser to communicate with their chatbot application. You can use this module

to create applications/bots that send and receive messages to the Cleverbot

chatbot application.

Google Translate: Googletrans is a free and unlimited python library that implemented Google Translate API. This uses the Google Translate Ajax API to make calls to such methods as detect and translate.

**2.1. Formation of functional and non-functional requirements**

|  |  |
| --- | --- |
| **Functional Requirements** | **Non-functional requirements** |
| Functional requirements help to understand the functions of the system. | They help to understand the system's performance. |
| Functional requirements are mandatory. | While non-functional requirements are not mandatory. |
| They are easy to define. | They are hard to define. |
| They describe what the product does. | They describe the working of product. |
| It concentrates on the user's requirement. | It concentrates on the expectation and experience of the user. |
| It helps us to verify the software's functionality. | It helps us to verify the software's performance. |
| These requirements are specified by the user. | These requirements are specified by the software developers, architects, and technical persons. |
| There is functional testing such as API testing, system, integration, etc. | There is non-functional testing such as usability, performance, stress, security, etc. |
| Examples of the functional requirements are - Authentication of a user on trying to log in to the system. | Examples of the non-functional requirements are - The background color of the screens should be light blue. |
| These requirements are important to system operation. | These are not always the important requirements, they may be desirable. |
| Completion of Functional requirements allows the system to perform, irrespective of meeting the non-functional requirements. | While system will not work only with non-functional requirements. |

**2.2 Applied Techniques for Chatbot**

**How Do Chatbots Work?**

Modern chatbot solutions leverage artificial intelligence, machine computer (ML) and natural language processing (NLP) to provide insights in real time. These technologies can instantly analyze data and suggest actions that are most likely to produce the desired result.

More than that though, AI chatbots are your website, social media, mobile app, etc, connected to a knowledge base and all connected data sources via NLP layers. communicates through channels such as

Natural language processing is the performance of chatbots to analyze and play human speech to provide contextually correct alerts.

NLP aims to make chatbot interactions feel like a conversation between two people, separating the two key times.

Natural language understanding (NLU) maintains a chatbot's human speech understanding data and is able to obtain it in a way that software can understand (purposes and entities).

Intents perform the users' goal, i.e. to find information or perform a certain action.

Entities complete the purpose and include dates, dimensions, location, etc. includes details such as

Natural language generation (NLG) is responsible for translating this data back into text.

The chatbot uses a dialect of XML known as Artificial Intelligence Markup Language (AIML) to identify patterns and metrics in its data before sending a response, so it can present responses contextually.

For example, if a customer enters a query in Messenger, the NLP layer first translates it into intents and entities. Following this, this behavioral data, usage history, etc. is analyzed to identify patterns that help provide a personal training with the best possible controls. runs with.

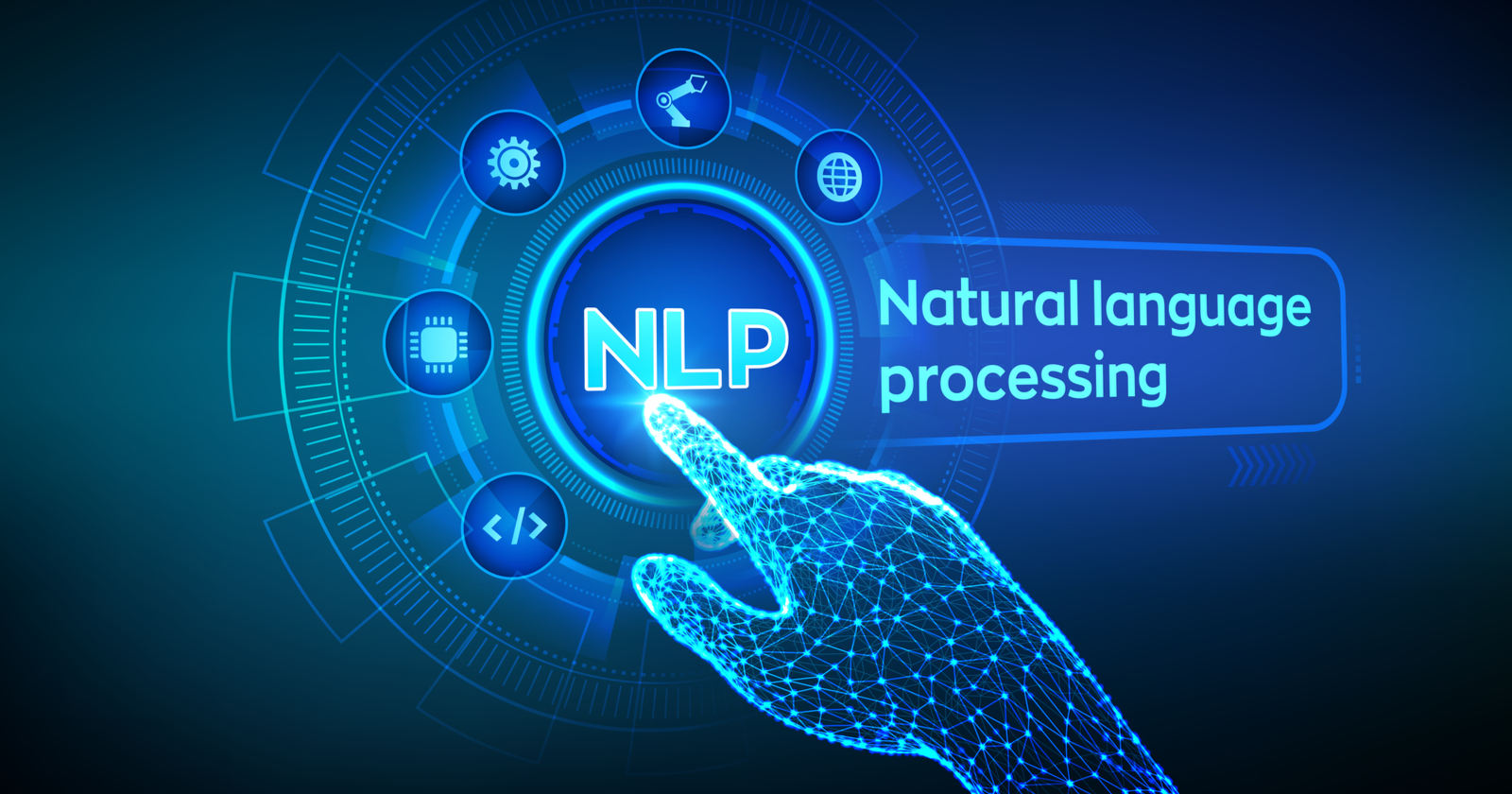
**2.2.1 Natural Language Processing**

***Natural Language Processing (NLP) Definition***

Natural language processing (NLP) is a branch of artificial intelligence (AI) that enables computers to comprehend, produce, and manipulate human language. Natural language processing has the ability to query data with correct language text or voice. This is also called "language input". Many consumers have probably interacted with NLP without even realizing it. For example, the core technology behind virtual assistants like Oracle Digital Assistant (ODA), Siri, Cortana or Alexa is NLP. When we ask these virtual assistants questions, it is NLP that allows the assistants to both understand the user's request and respond in natural language. NLP applies to both written and spoken text and can be applied to all languages. Other examples of NLP-powered tools include web search, spam filters, automatic text or speech translation, document summarization, sentiment analysis, and grammar/spelling. For example, some e-mail programs may suggest appropriate replies based on message content. These programs use NLP technology to read, analyze and respond to your message.

There are several other terms that are used synonymously with NLP in a general sense. Natural language understanding (NLU) means understanding human language using computers, and natural language generation (NLG) means producing. NLG has the feature of verbal explanation of a situation. This is also called "language output", which summarizes and converts meaningful information into text with the help of a concept known as "graphic grammar".

Figure:NLP



In practice, NLU is used to mean NLP. It is the understanding that allows developers and users to interact with computers with natural sentences and communications by understanding the structure and meaning of computers, all languages. Computational linguistics (CL) is a scientific field that studies the computational properties of human languages. NLP, on the other hand, is an engineering discipline that deals with building computerized entities that understand, generate, or manipulate human language.

NLP studies began shortly after the invention of digital computers in the 1950s, and NLP draws on both linguistics and artificial intelligence. But major breakthroughs in the last few years have come from machine learning, a branch of artificial intelligence that develops systems that learn from data and make generalizations. Deep learning is a variant of machine learning that can learn very complex patterns from large data sets. So it's ideal for learning natural language complexities from web-sourced datasets.

**NLP Programming Languages**

Python:

NLP Libraries and toolkits are usually available in Python. This is because the vast majority of NLP projects are developed in Python. Python's interactive development environment makes it easy to develop and test new code.

Java and C++:

C++ and Java are generally preferred for handling large amounts of data because of their more efficient code support.

**2.2.2 Pattern Recognition (Matching, Mapping)**

"Pattern recognition" is the process of recognizing patterns, regularities, or relationships in data or information. It involves identifying and categorizing patterns, as well as detecting and describing relationships and dependencies between variables.

There are two main types of pattern recognition: matching and mapping.

"Matching" is the process of finding a pattern or a set of patterns in an input and comparing it to a set of stored patterns to determine the closest match. This is often used in image recognition, where an image is compared to a set of stored images to determine what object it depicts.

"Mapping" is the process of creating a map or a representation of the relationship between two sets of variables. This can be used to describe patterns in data, as well as to create predictions based on past observations.

Both of these techniques are widely used in a variety of applications, including machine learning, artificial intelligence, computer vision, and data analysis.

Figure:Pattern Recognition



**2.2.3 Semantic web**The "Semantic Web" is an idea proposed by Tim Berners-Lee, the inventor of the World Wide Web. It refers to a vision of the Web where information is not only connected and accessible, but also meaningful and understandable by computers.

The Semantic Web is built on the concept of linking data and making it more machine-readable, so that computers can understand the relationships between different pieces of information and use this understanding to perform tasks such as data integration, reasoning, and decision making.

To achieve this, the Semantic Web uses technologies such as RDF (Resource Description Framework), OWL (Web Ontology Language), and SPARQL (SPARQL Protocol and RDF Query Language) to represent and link data in a machine-readable format. This allows computers to automatically process and analyze vast amounts of data, enabling new applications and services that were not possible before.

The Semantic Web is still an evolving field, and its full potential has yet to be realized, but it holds promise for transforming the Web into a more intelligent and interconnected space.

Figure:Semantic Web



**2.2.4 Data Mining**

"Data mining" is the process of discovering patterns, relationships, and insights in large amounts of data. It involves the use of statistical and machine learning techniques to analyze and extract information from data, often with the goal of making better decisions and identifying new opportunities.

Data mining is used in a variety of industries and applications, including marketing, finance, healthcare, and scientific research. It can help organizations identify patterns and trends in customer behavior, detect fraud, improve customer relationships, and make more informed decisions.

Data mining typically involves several steps, including data preparation, pattern discovery, pattern evaluation, and knowledge representation. These steps help ensure that the information obtained from the data is accurate, meaningful, and useful.

Overall, data mining is an important tool for organizations that want to leverage their data to make better decisions and gain a competitive advantage.

Figure:Data mining

**2.3 Architecture of Chatbot**

The architecture of a chatbot can vary depending on the specific requirements and objectives of the chatbot, but typically includes the following components:

Natural Language Processing (NLP): This component is responsible for understanding and processing the natural language used by users to interact with the chatbot. NLP techniques such as tokenization, stemming, and named entity recognition are used to extract meaningful information from user input.

Dialogue Management: This component is responsible for managing the flow of the conversation between the user and the chatbot. It determines the appropriate response to a user's input based on the context of the conversation and the information available.

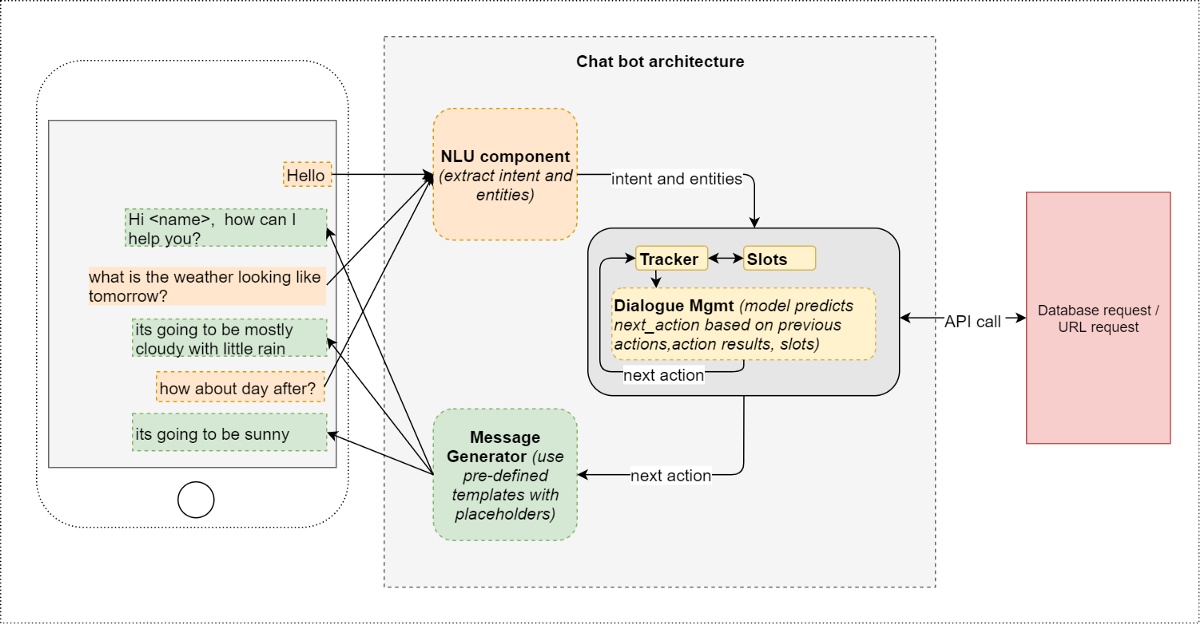
Knowledge Base: This component stores the information and data that the chatbot uses to generate its responses. It may include databases, APIs, or other sources of information that the chatbot can access to provide relevant and accurate answers to users.

User Interface: This component is responsible for presenting the chatbot's responses to the user and allowing users to interact with the chatbot. This may include a graphical interface, text-based interface, or voice-based interface, depending on the chatbot's design and capabilities.

Analytics and Reporting: This component is responsible for collecting and analyzing data on user interactions with the chatbot. This information can be used to improve the performance and accuracy of the chatbot over time.

Overall, these components work together to provide a seamless and intuitive experience for users interacting with the chatbot, allowing them to easily access information and complete tasks.

Figure:Architecture of Chatbot



**2.4 Choosing a chatbot development platform**

When choosing a chatbot development platform, there are several factors to consider, including:

Functionality: Consider the features and capabilities offered by the platform, such as NLP, integration with external systems, and the ability to customize the chatbot's appearance and behavior.

Ease of Use: Look for a platform that is user-friendly and does not require extensive technical expertise to use. This can help speed up development and reduce costs.

Scalability: Consider the platform's ability to scale to meet the growing demands of your chatbot. Can it handle increasing traffic and data volumes, and does it provide the necessary resources to ensure reliable performance?

Integrations: Evaluate the platform's ability to integrate with other systems and technologies, such as databases, APIs, and messaging platforms.

Pricing: Consider the cost of the platform, including any ongoing fees and usage charges. Be sure to compare pricing and features across multiple platforms to find the best value for your budget.

Support and Documentation: Look for a platform that provides comprehensive support and documentation to help you get started and troubleshoot any issues that may arise.

Community and Resources: Consider the size and activity level of the platform's community and the availability of resources, such as forums, documentation, and tutorials. This can help you get answers to questions and find support when needed.

By considering these factors, you can choose a chatbot development platform that best meets your specific needs and requirements.

**3.Design and Implementation of AI chatbot**

. Here are the steps to follow:

Define the purpose and scope of the chatbot: Before starting the design, you need to define the purpose and scope of the chatbot. For example, the chatbot can be used to answer employee questions about company policies, benefits, or provide assistance with onboarding.

Determine the platform and programming language: Once you have defined the purpose and scope, you need to determine the platform and programming language to use for the chatbot. There are various platforms available such as Dialogflow, IBM Watson, and Amazon Lex. You can also use programming languages such as Python, Java, or Node.js.

Create a conversational flow: After selecting the platform and programming language, create a conversational flow that maps out the possible interactions between the chatbot and the user. This should include the possible questions or statements that the user might make and the corresponding responses from the chatbot.

Develop the chatbot: Once you have created the conversational flow, you can start developing the chatbot. This involves coding the conversational flow into the chosen platform or programming language. You may also need to integrate the chatbot with other systems, such as HR databases or knowledge management systems.

Train the chatbot: To improve the chatbot's accuracy and effectiveness, you need to train it using a combination of supervised and unsupervised learning techniques. This involves providing the chatbot with sample questions and answers, and then using natural language processing algorithms to identify patterns and improve its responses.

Test and refine the chatbot: Once the chatbot has been developed and trained, it's important to test it thoroughly and refine it based on user feedback. This involves conducting usability tests, analyzing user feedback, and making improvements to the conversational flow and the chatbot's responses.

Deploy the chatbot: After testing and refining the chatbot, you can deploy it to your HR platform or website, making it available to employees.

Overall, designing and implementing an AI chatbot for HR requires a combination of technical expertise and understanding of HR policies and procedures. However, the benefits of having an AI chatbot for HR can be significant, including increased efficiency, improved accuracy, and enhanced employee satisfaction.

**3.1 Design of AI chatbot**

Here is a high-level overview of the design of an AI chatbot:

Identify the purpose and scope: The first step in designing an AI chatbot is to identify its purpose and scope. What problem does it solve? Who is the target audience? What kind of tasks or questions should the chatbot be able to handle?

Choose the platform and technology stack: Once you have a clear understanding of the chatbot's purpose and scope, you need to choose the platform and technology stack to use. There are many chatbot platforms available, including Dialogflow, IBM Watson, and Amazon Lex. You also need to choose the programming language and frameworks you will use to build and train your chatbot.

Define the conversational flow: Next, you need to define the conversational flow of the chatbot. This involves mapping out all the possible interactions that a user might have with the chatbot and the corresponding responses. You can use a tool like a flowchart or decision tree to visualize the conversational flow.

Develop and train the chatbot: With the conversational flow defined, you can begin developing and training the chatbot. This involves coding the conversational flow into the chosen platform or programming language and training the chatbot with appropriate data. You may also need to integrate the chatbot with other systems, such as databases or APIs.

Test and refine the chatbot: Once the chatbot has been developed and trained, you need to test it to ensure it is functioning as intended. You can use automated testing tools or perform manual testing to identify bugs or areas for improvement. Based on feedback from testing, you can refine the conversational flow and training data to improve the chatbot's performance.

Deploy the chatbot: Once you are satisfied with the chatbot's performance, you can deploy it to your desired platform or channels, such as a website or messaging app.

Monitor and maintain the chatbot: After deploying the chatbot, you need to monitor its performance and user feedback. This can help identify areas for further improvement or additional functionality. You may also need to maintain the chatbot by updating training data or fixing bugs that may arise over time.

Overall, designing an AI chatbot requires a combination of technical expertise, user research, and a deep understanding of the chatbot's purpose and audience. By following these steps, you can create an AI chatbot that meets your users' needs and delivers an engaging and efficient user experience.

3.2 Software implementation of AI chatbot

Figure: Chatbot with results and input

Figure: Test code with results and input

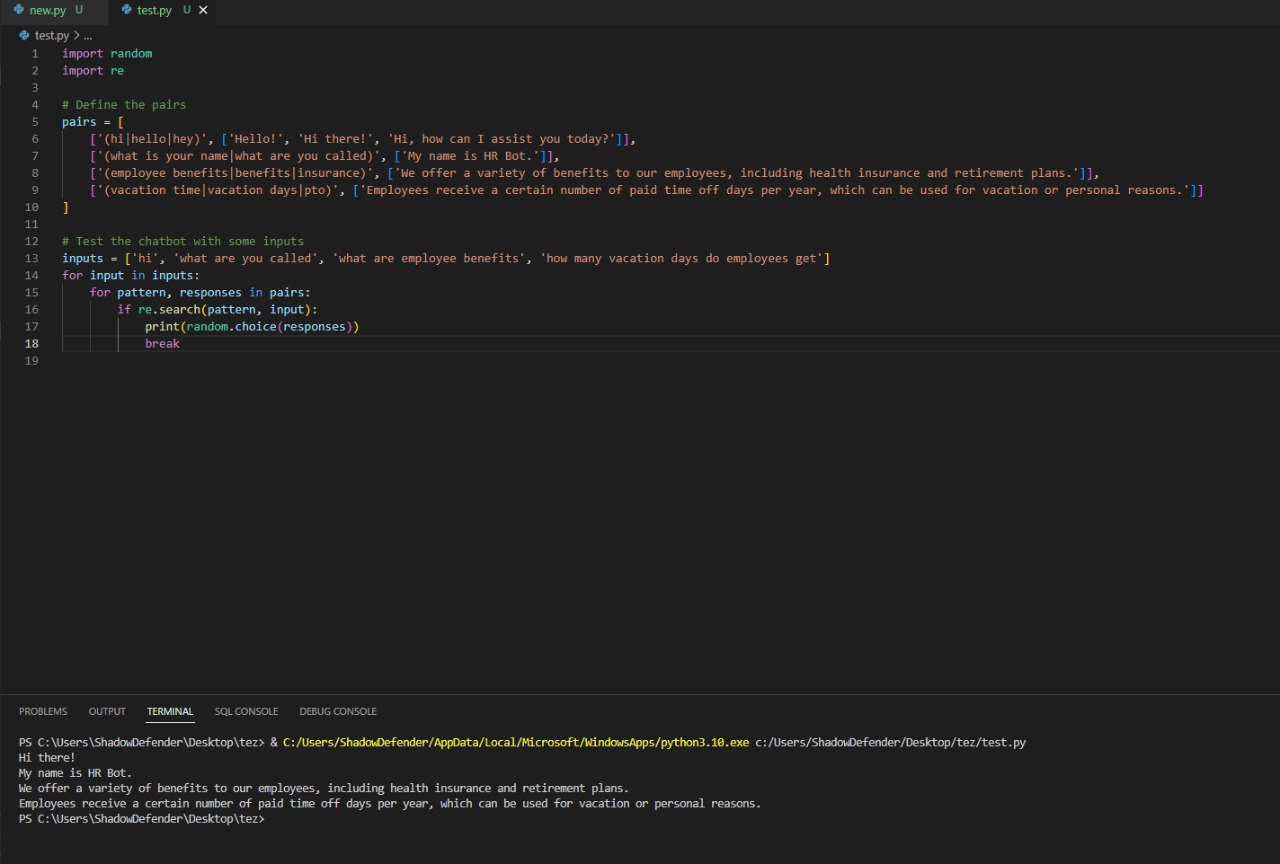
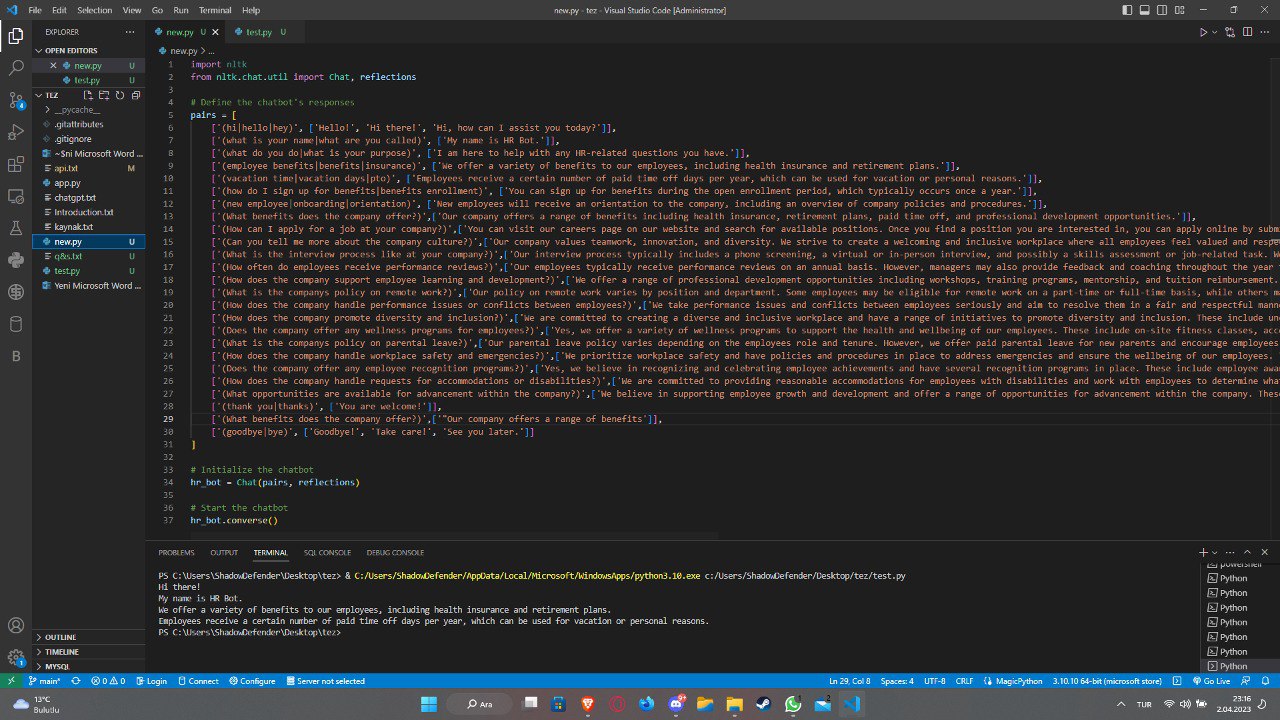


Figure:Diagram(Test case)

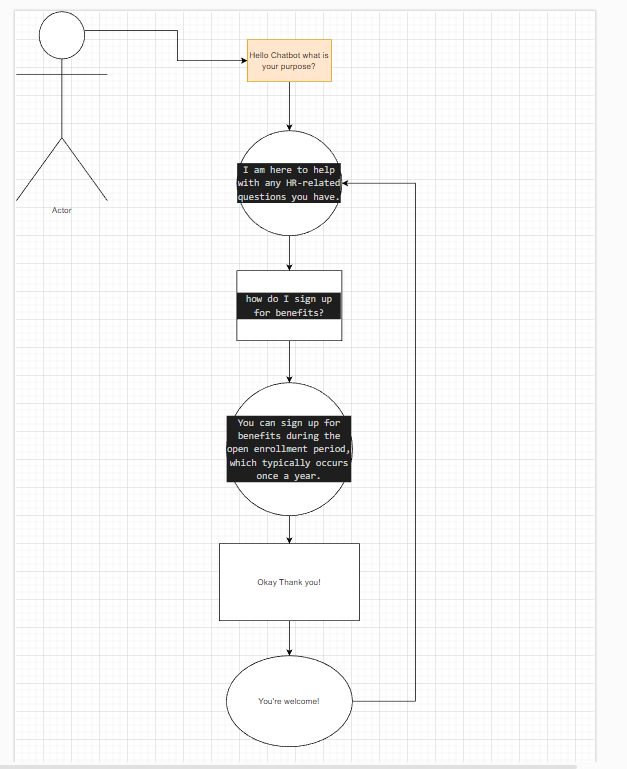


Figure:Diagram(Test Case)

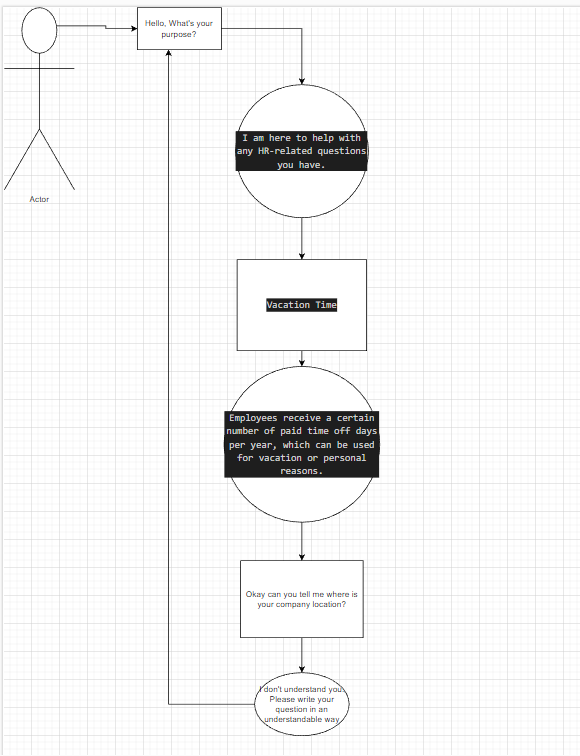


Figure:Diagram(Test Case)

