**Building a Chatbot from Scratch using Rasa Framework**

**Understanding Rasa**

**Rasa is an open-source framework for building conversational AI applications, which includes tools for Natural Language Understanding (NLU) and dialogue management. It allows developers to create intelligent chatbots that can understand user inputs, respond appropriately, and learn from interactions.**

**Installation and Setup**

**To begin building a chatbot with Rasa, you'll first need to install Python on your machine. Once Python is installed, you can proceed with installing Rasa.**

1. **Install Rasa  
   Run the following command to install Rasa:**

pip install rasa  
  
 **2. Create a New Rasa Project  
 After Rasa is installed, create a new project directory and initialize a new Rasa project by running the following command:**

rasa init

This will prompt you to select a default directory to create the Rasa model. You can use the default directory for now.

1. **Training the Model**  
   Once the project is initialized, you can train the Rasa model by running:

**rasa train**

This will train the model based on the default configurations and datasets provided.

1. **Run the Rasa Server**  
   To start the Rasa server, use the following command:

**rasa run**

You can also test the chatbot by interacting with it in the Python shell or terminal.

**Training the NLU Model**

Training the Natural Language Understanding (NLU) model is essential for enabling the chatbot to understand user intents and extract entities accurately. Here’s how you can train the NLU model:

1. **Train the NLU Model**  
   Run the following command to train the NLU model:

rasa train nlu

This will train the model on your custom NLU data, which includes examples of user intents and entity labels.

1. **Test the Model**  
   After training, you can test the NLU model to check if it recognizes the intents and entities correctly according to your project’s requirements.

**Deployment**

After completing the training and testing processes, you can deploy your chatbot to a production environment. This enables real users to interact with the chatbot and derive value from its capabilities. Deployment can involve hosting the chatbot on a web server, integrating it with messaging platforms (such as Slack, Facebook Messenger, or custom websites), and setting up continuous updates for improving its performance.