**WEEK – 01 (Milestones)**

**Week 1 Project Milestone** in implementing a chatbot for an NLP project, here's how you can structure the milestone and its implementation steps:

**Project Milestone: Implementing Chatbot Core Features**

**Objective:**

Develop the foundational components of an NLP-based chatbot capable of understanding user inputs, responding appropriately, and providing basic conversational interactions.

**Deliverables:**

1. **Chatbot Framework Setup**
   * Select a chatbot framework/library (e.g., Rasa, Dialogflow, ChatterBot, or custom Python solution using NLP libraries like spaCy, Hugging Face, or NLTK).
   * Set up the development environment (e.g., Python, required libraries).
2. **Natural Language Understanding (NLU) Pipeline**
   * Implement a pipeline for:
     + Text preprocessing (tokenization, stemming/lemmatization, removing stop words).
     + Intent recognition (using pre-trained models or custom classifiers).
     + Entity extraction (e.g., names, dates, or specific keywords).
3. **Basic Conversation Flow**
   * Create a rule-based or retrieval-based conversation flow for common queries.
   * Define intents and responses (e.g., greetings, FAQs, or fallback responses).
4. **Integration with Input/Output Interface**
   * Set up a simple UI or command-line interface (CLI) to send user queries and receive chatbot responses.
5. **Testing and Validation**
   * Conduct basic testing to ensure the chatbot correctly identifies intents and responds as expected.
   * Log interactions for debugging and improving responses.

**Steps for Implementation:**

**Step 1: Framework Setup**

* Install dependencies:
* pip install rasa spacy transformers NLTK
* Set up the project folder structure:
* /chatbot-project
* ├── data/
* ├── models/
* ├── scripts/
* ├── logs/
* └── main.py

**Step 2: NLU Pipeline**

* Define intents in nlu.yml (if using Rasa) or a custom JSON structure:
* - intent: greet
* examples: |
* - hello
* - hi
* - good morning
* - hey
* Train the NLU model:
* rasa train

**Step 3: Conversation Flow**

* For rule-based responses, define dialogue rules in domain.yml:
* responses:
* utter\_greet:
* - text: "Hello! How can I help you today?"
* For retrieval-based, use embeddings with Hugging Face’s Sentence Transformers:
* from sentence\_transformers import SentenceTransformer, util
* model = SentenceTransformer('paraphrase-MiniLM-L6-v2')

**Step 4: Interface**

* Use a CLI or Flask/Django for the interface:
* from flask import Flask, request, jsonify
* app = Flask(\_\_name\_\_)
* @app.route('/chat', methods=['POST'])
* def chat():
* user\_input = request.json['message']
* response = get\_response(user\_input) # Implement this function
* return jsonify({'response': response})

**Step 5: Testing**

* Test inputs like:
* User: Hi
* Bot: Hello! How can I help you today?

**Milestone Success Criteria:**

* The chatbot can:
  + Understand at least 3-5 intents (e.g., greetings, small talk, basic queries).
  + Respond appropriately to user inputs.
  + Log interactions for improvement.
* A simple UI or CLI is ready for testing interactions.

**Future Steps (Post Week 1):**

1. Add advanced NLU features (contextual understanding, multi-turn conversations).
2. Incorporate external API integrations (e.g., weather, news).
3. Deploy on platforms (e.g., Telegram, WhatsApp, or Web).

Would you like detailed code examples or architecture diagrams for any part?