Company Details Extraction Chatbot

Project overview:

**Title:** Company Details Extraction ChatBot

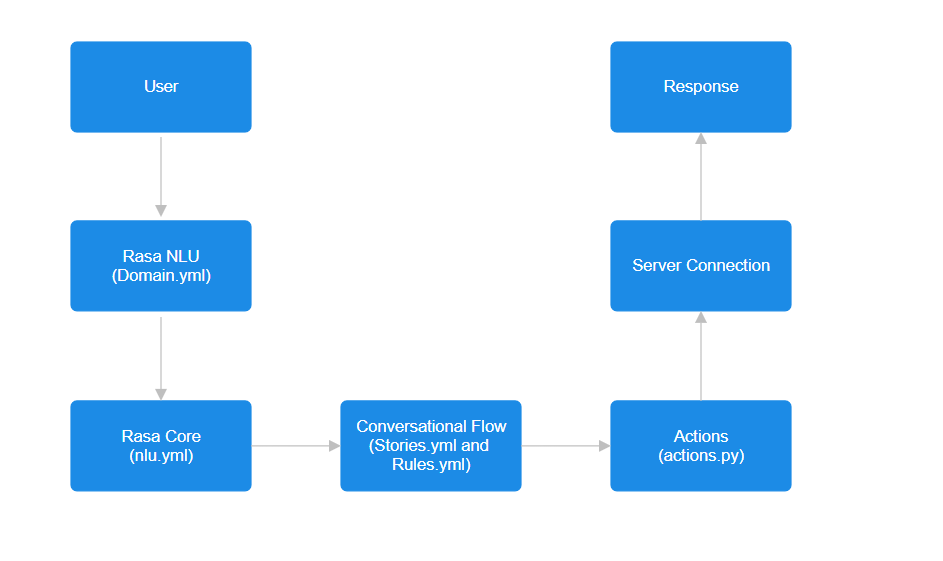
**Overview:**

The chatbot is designed to interact with users and give basic information about companies. Data such as the company website, metadata, and social media links are extracted through web scraping techniques. The bot is implemented using Rasa, an open-source conversational AI framework.

**Objective:**

1. Enable users to greet and say goodbye in a friendly and conversational manner.
2. Let users request information about companies by mentioning their names.
3. The application dynamically scrapes company websites, extracting metadata and contact details such as social media.

System Architecture:



**Components:**

**1)Domain Configuration (domain.yml):**

Describes intents, entities, responses, and actions.

Controls session settings for an interaction.

**2)Natural Language Understanding (nlu.yml):**

Includes training data for intents (greet, goodbye, scrape\_company) and examples for entity recognition (company\_name).

**3)Conversation Flow (stories.yml and rules.yml):**

*Stories:*

Greet the user, process queries about company information, and goodbye.

*Rules:*

Handle particular cases like a greeting or questions about a company.

**4) Actions (actions.py):**

i)The action\_scrape\_company finds the website of the company.

Extracts metadata, contact details, and social media links using BeautifulSoup and requests.

ii)Answers the dynamic query of users with the data scraped.

Project Setup:

**Environment Setup:**

*pip install rasa[full]*

**Project initialization:**

*rasa init --no-prompt*

**Description of Folders:**

/actions - Custom actions folder

/data - Training data (NLU and stories)

/models - Trained models

config.yml - NLU and policy configuration

domain.yml - Intents, responses, entities, slots, and actions

endpoints.yml - Endpoint configurations for action servers

credentials.yml - Channel credentials

**Technology Stack:**

**Framework:**  Rasa (version 3.6.21)

**Libraries:**

1. rasa\_sdk : For custom actions
2. BeautifulSoup4 : For HTML parsing
3. Requests : for HTTP requests

**Languages:** python(version 3.10)

**Library installation**

1. **rasa\_sdk:** *pip install rasa\_sdk*
2. **Beautifulsoup4:** *pip install beautifulsoup4*
3. **Requests:** *pip install requests*

Implementation Details:

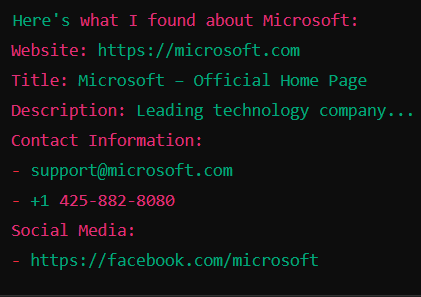
1. **Web Scraping:**
2. Company Website URLS are generated based on common domain variations
3. Metadata and social links are extracted from HTML content
4. **Error Handling:**
5. Handles errors like Invalid company names, website not found and HTTP exceptions gracefully, providing appropriate feedback to the user

Usage WorkFlow:

1. **User Input:**
2. User types a message like: “Tell me about Microsoft”
3. **Intent and Entity Detection:**
4. The bot identifies ***scrape\_company*** intent and extracts the ***company\_name*** entity
5. **Custom Action execution:**
6. The ***action\_scrape\_company*** action:

1. Find the company website
2. Scrapes and process the website to extract metadata, contact info and social links
3. **Bot response:**

The bot provides the structured response with scraped information



Testing And Validation:

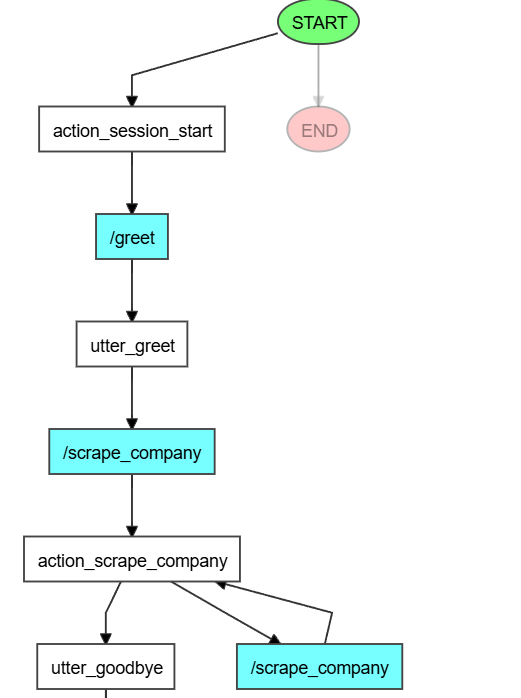
1. **Integration Testing:**

Test the end-to-end conversational flow should be tested to ensure seamless experience

1. **Interactive Testing:**

To test the chatbot more interactive

***Command:*** *rasa interactive*

**

1. **Command Testing:**

*To test the chatbot generally*

***Command:*** *rasa test*

1. **Error Scenarios:**

Validate error handling when:

1. no company name is provided
2. A company website is not Found
3. Scraping encounters unexpected issues

Deployment:

1. **Train the Chatbot:**

To train the chatbot

***Command:*** rasa train

1. **Run the action server:**

To run the server:

***Command:*** *rasa run actions*

1. **To start the action**

To start interact with actions

***Command:*** *rasa shell*

To start interact in API server

***Command:*** *rasa run --enable-api --cors “\*”*