

Building a Conversational Agent for Bank Account Creation with Rasa

Introduction

The goal of this project was to develop a simple conversational agent capable of guiding users through the process of creating a bank account. The project utilised the Rasa platform, an open-source natural language processing (NLP) tool designed for building conversational AI.

Overview of Conversational Agent Design

Intent Identification

The first step in designing the conversational agent was to identify the key intents or user goals. These intents included greetings, goodbyes, inquiries about account types, identification documents, initial deposit, and requests to talk to a human.

Entity Recognition

Entities such as `account_type`, `identification_document`, `initial_deposit`, `name`, and `phone_number` were identified to extract relevant information from user messages. These entities for now are not that useful, but can be for future expansion of a project and can play a crucial role in personalising responses and guiding the user through the account creation process.

Dialogue Management

To manage the flow of conversations, rules and stories were defined. Rules captured simple interactions, while stories represented more complex dialogues. The `talk_to_human` intent triggered a rule to connect the user to a human representative.

Responses and Utterances

Responses were crafted to provide informative and helpful answers to user queries. Dynamic responses, such as `utter_contact_info`, were designed to incorporate extracted entity values like the user's name and the phone number.

Implementation Steps

1. Project Setup

The project structure followed Rasa conventions, with organised directories for data, domain, actions, and configuration files.

2. NLU Training Data

Natural Language Understanding (NLU) training data (in `nlu.yml`) contained examples for each intent, including variations and expressions likely to be used by users. Entities were annotated within training examples to train the model to recognize them.

3. Domain Configuration

The `domain.yml` file defined the structure of the conversational agent. It included intents, responses, entities, and slots. Dynamic responses, like `utter_contact_info`, were configured to use entity values such as the user's name and phone number.

4. Dialogue Management

Rules in `rules.yml` captured simple, predictable interactions, while stories in `stories.yml` represented more complex and diverse user journeys. These stories helped the model learn the correct responses for various scenarios.

5. Model Training

The model was trained using the `rasa train` command, incorporating the annotated NLU data, stories, and rules. Training iterations refined the model's ability to understand user inputs and generate appropriate responses.

6. Testing and Iteration

The conversational agent was tested using the Rasa shell, allowing for interactive conversations. Iterative testing and refinement were performed to enhance the agent's understanding, handling of entities, and overall performance.

Principles and Techniques Applied

1. Entity-Centric Design

The design focused on identifying and extracting entities crucial to the account creation process. This entity-centric approach allowed for dynamic and personalised responses based on user-provided information.

2. Modular Architecture

The modular architecture of the Rasa project facilitated the organisation of components such as intents, entities, responses, and actions. This modularity enhanced code readability and maintainability.

3. Intent-Entity Pairing

Intents and entities were paired logically to capture user intentions and the specific information needed for account creation. For example, the `create_account` intent often co-occurred with `account_type` and `identification_document` entities and followed with the name and phone number of a user.

4. Rule-Based Dialogue Management

Simple and deterministic interactions, such as greetings and goodbyes, were captured using rules. This approach ensured precise handling of expected user inputs.

Challenges and Learnings

During the project, several challenges were encountered and addressed. These challenges included refining entity recognition, ensuring diverse and robust training data, and optimising the handling of user queries with dynamic responses.

The iterative development process, involving continuous testing and refinement, was key to overcoming these challenges. User feedback, especially during interactive testing, played a crucial role in identifying areas for improvement.

Results

The conversational agent was subjected to rigorous testing to evaluate its performance and effectiveness in handling various user scenarios. Three distinct conversations were simulated to assess the chatbot's capability to guide users through the bank account creation process and handle different user requests.

Scenario 1: Information Gathering and Phone Call Request

In the first scenario, a user expressed the intent to gather all the necessary information for creating a bank account and then got a phone call from a manager to proceed with the account creation process. The conversational agent successfully guided the user through the steps, provided relevant information, and acknowledged the user's request for a manager's call.

Scenario 2: Refusal to Engage with Chatbot and Manager Connection Request

In the second scenario, the user explicitly refused to interact with the chatbot and insisted on connecting to a manager. The user provided both a name and a phone number, expressing a clear preference for a phone call from a human representative. The conversational agent successfully recognized the user's request and triggered the appropriate action to connect the user to a manager.

Scenario 3: General Information Inquiry without Phone Call Request

In the third scenario, the user expressed a general interest in obtaining information without the intention of receiving a phone call from a manager. The conversational agent provided the necessary details regarding the account creation process without initiating a manager connection.

```
Bot loaded. Type a message and press enter (use '/stop' to exit):
Your input -> hello
Hi! How can I help you?
Your input -> i'd like to create an account
To create a new bank account, you'll need to follow these steps:
1. Choose the type of account you want to open.
2. Gather your identification documents.
3. Visit our nearest branch.
4. Provide the necessary information and make the initial deposit.
5. Congratulations! Your account is now open.
Please provide your contact information and our manager will get in touch with you shortly.
Or would you like to know anything else?
Your input -> do i need any documents to open an account?
You'll need to provide a valid ID proof to open a bank account. You can use any of the following documents: Passport, ID, Driver's License, etc.
Your input -> How much money do I need to open a bank account?
You'll need to make an initial deposit minimum of $60 to open a bank account.
Your input -> ok, i'm ready. my name is Iskander, my phone number is +79991234567
Thank you! Our customer service representative will get in touch with you shortly.
Your input -> thank you, bye!
Hope I was able to help you. Bye!
Your input -> █
```

```
Bot loaded. Type a message and press enter (use '/stop' to exit):
Your input -> who are you?
I am a bot to guide you through the process of creating a bank account.
What would you like to know?
Your input -> i don't want to talk to you connect me with a real human
Please provide your contact information and our manager will get in touch with you shortly.
Your input -> my name is Iskander, the number +71234567890
Thank you! Our customer service representative will get in touch with you shortly.
Your input -> thank you, bye
Hope I was able to help you. Bye!
Your input -> █
```

```
Bot loaded. Type a message and press enter (use '/stop' to exit):
Your input -> hello
Hi! How can I help you?
Your input -> who are you?
I am a bot to guide you through the process of creating a bank account.
What would you like to know?
Your input -> i want to create a bank account. how can i do it?
To create a new bank account, you'll need to follow these steps:
1. Choose the type of account you want to open.
2. Gather your identification documents.
3. Visit our nearest branch.
4. Provide the necessary information and make the initial deposit.
5. Congratulations! Your account is now open.
Please provide your contact information and our manager will get in touch with you shortly.
Or would you like to know anything else?
Your input -> what types of account are available?
We offer two types of bank accounts: Savings Account and Checking Account.
Which one would you like to open?
Your input -> i want to know if it's possible to use my driver's license?
You'll need to provide a valid ID proof to open a bank account. You can use any of the following documents: Passport, ID, Driver's License, etc.
Your input -> ok, thanks, bye!
Hope I was able to help you. Bye!
Your input -> █
```

Conclusion

In conclusion, the project successfully implemented a conversational agent for bank account creation using the Rasa platform. By adhering to principles such as entity-centric design, modular architecture, and dynamic responses, the agent demonstrated effective handling of user queries and provided personalised assistance.

The iterative development process, from project setup to model training and testing, allowed for continuous improvement and refinement. The principles and techniques applied can serve as a foundation for building more sophisticated conversational agents and adapting them to various domains and user needs.

References:

Rasa Documentation:

Authors: Rasa Technologies GmbH

URL: <https://rasa.com/docs/>

Dialogue Natural Language Understanding Challenges and Recent Trends:

Authors: Anusha Balakrishnan, Mahmoud Azab, Alan W. Black, Joao Sedoc, Chris Falter, and Kevin P. Gimpel

URL: <https://www.aclweb.org/anthology/2020.acl-main.259/>

Conversational Agents: Goals, Technologies, Vision and Challenges:

Authors: Merav Allouch, Amos Azaria, and Rina Azoulay

URL: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC8704682/>

HAIID Lectures and Labs materials