

Building a chatbot using a Recurrent Neural Network (RNN) in Python

□ **Step 1: Data Preparation**

Prepare a dataset with conversation pairs. To create a chatbot that provides information about fruits.

```
conversations = [  
    ("Tell me about apples.", "Apples are one of the  
    most popular fruits in the world."),  
    ("What's the color of a banana?", "Bananas are  
    typically yellow when ripe."),  
    ("How can I store grapes?", "Grapes should be  
    stored in the refrigerator."),  
    # Add more conversation pairs  
]
```

□ **Step 2: Data Preprocessing**

In this step, Tokenize the text data and create input and target sequences for training.

```
import tensorflow as tf  
from tensorflow.keras.preprocessing.text import  
Tokenizer  
from tensorflow.keras.preprocessing.sequence  
import pad_sequences  
  
# Tokenize the text data  
tokenizer = Tokenizer()  
tokenizer.fit_on_texts(conversations)
```

```

total_words = len(tokenizer.word_index) + 1

# Create input and target sequences
input_sequences = []
target_sequences = []

for conversation in conversations:
    input_text, target_text = conversation
    input_seq =
tokenizer.texts_to_sequences([input_text])[0]
    target_seq =
tokenizer.texts_to_sequences([target_text])[0]

    input_sequences.append(input_seq)
    target_sequences.append(target_seq)

max_sequence_length = max([len(seq) for seq in
input_sequences])
input_sequences =
pad_sequences(input_sequences,
maxlen=max_sequence_length, padding='post')
target_sequences =
pad_sequences(target_sequences,
maxlen=max_sequence_length, padding='post')

```

☐ **Step 3:** Build the RNN Model

Create an RNN model using Keras as before. To use a simple RNN-based sequence-to-sequence model.

```

from tensorflow.keras.models import Sequential

```

```
from tensorflow.keras.layers import Embedding,  
SimpleRNN, Dense
```

```
model = Sequential()  
model.add(Embedding(total_words, 100,  
input_length=max_sequence_length))  
model.add(SimpleRNN(100,  
return_sequences=True))  
model.add(Dense(total_words,  
activation='softmax'))
```

```
model.compile(optimizer='adam',  
loss='categorical_crossentropy')
```

☐ **Step 4:** Training the Model

Train the RNN model on a dataset.

```
from tensorflow.keras.utils import to_categorical
```

```
X = input_sequences  
Y = to_categorical(target_sequences,  
num_classes=total_words)
```

```
model.fit(X, Y, epochs=100, verbose=1)
```

☐ **Step 5:** Generating Responses

Create a function to generate responses based on user input.

```
def generate_response(input_text):
```

```
    input_seq =
tokenizer.texts_to_sequences([input_text])[0]
    input_seq = pad_sequences([input_seq],
maxlen=max_sequence_length, padding='post')
    predicted_word_index =
model.predict_classes(input_seq, verbose=0)[0]
    predicted_word =
tokenizer.index_word[predicted_word_index] if
predicted_word_index > 0 else "
    return predicted_word
```

Example conversation loop

while True:

```
    user_input = input("You: ")
```

```
    if user_input.lower() == 'exit':
```

```
        print("Chatbot: Goodbye!")
```

```
        break
```

```
    response = generate_response(user_input)
```

```
    print("Chatbot:", response)
```

Sample Output:

You: Tell me about apples.

Chatbot: Apples are one of the most popular fruits in the world.

You: What's the color of a banana?

Chatbot: Bananas are typically yellow when ripe.

You: How can I store grapes?

Chatbot: Grapes should be stored in the refrigerator.

You: exit

Chatbot: Goodbye!