Here are some common feature extraction techniques for chatbots:

1.Tokenization: *Break input text into individual words or tokens. This allows the chatbot to analyze and process user messages on a word-by-word basis.*

2. Part-of-Speech (POS) Tagging: *Assign grammatical parts of speech (e.g., noun, verb, adjective) to each word in a sentence. This helps in understanding the structure of the input.*

3.med Entity Recognition (NER):*Identify and classify entities such as names of people, places, organizations, and dates in the text. NER is crucial for handling user queries related to specific entities.*

4.N-grams: *Extract sequences of adjacent words (bigrams, trigrams, etc.) to capture context and phrases within the input text.*

5.TF-IDF (Term Frequency-Inverse Document Frequency):*Calculate the importance of words in a document relative to a collection of documents. This can help in identifying key terms in user queries.*

Train a machine learning model:

The next step is to train a machine learning model. We'll use the processed data to train a neural network using the TensorFlow library. Here's the code to train the model:

```
Import tensorflow as tf
From tensorflow.keras.preprocessing.text import Tokenizer
```

```
From tensorflow.keras.preprocessing.sequence import pad_sequences

# Set parameters
Vocab_size = 5000
Embedding_dim = 64
Max_length = 100
Trunc_type='post'
Padding_type='post'
Oov_tok = "<OOV>"
Training_size = len(processed_data)

# Create tokenizer
Tokenizer = Tokenizer(num_words=vocab_size, oov_token=oov_tok)
Tokenizer.fit_on_texts(processed_data)
```

```
Word_index = tokenizer.word_index

# Create sequences
Sequences =
tokenizer.texts_to_sequences(processed_data)
Padded_sequences =
pad_sequences(sequences,
maxlen=max_length, padding=padding_type,
truncating=trunc_type)

# Create training data
Training_data =
padded_sequences[:training_size]
Training_labels =
padded_sequences[:training_size]

# Compile model
```

```
Model.compile(loss='sparse_categorical_crossentropy', optimizer='adam', metrics=['accuracy'])
```

```
# Train model
Num_epochs = 50
History = model.fit(training_data, training_labels, epochs=num_epochs, verbose=2)
```

Evaluation:

We have built a simple chatbot using Python and TensorFlow. We started by gathering and preprocessing data, then we built a neural network model using the Keras Sequential API. We then created a simple command-line interface for the chatbot and tested it with some example conversations.

This is just a basic example of a chatbot, and there are many ways to improve it. With more advanced techniques and tools, you can build chatbots that can understand natural language, generate human-like responses, and even learn from user interactions to improve over time.

Submitted by

Jabez
Raja
Manimaran
Antony Raja
Abiman