

A solid blue abstract shape in the top-left corner, resembling a stylized 'L' or a corner element.

# AI Chatbot Creation

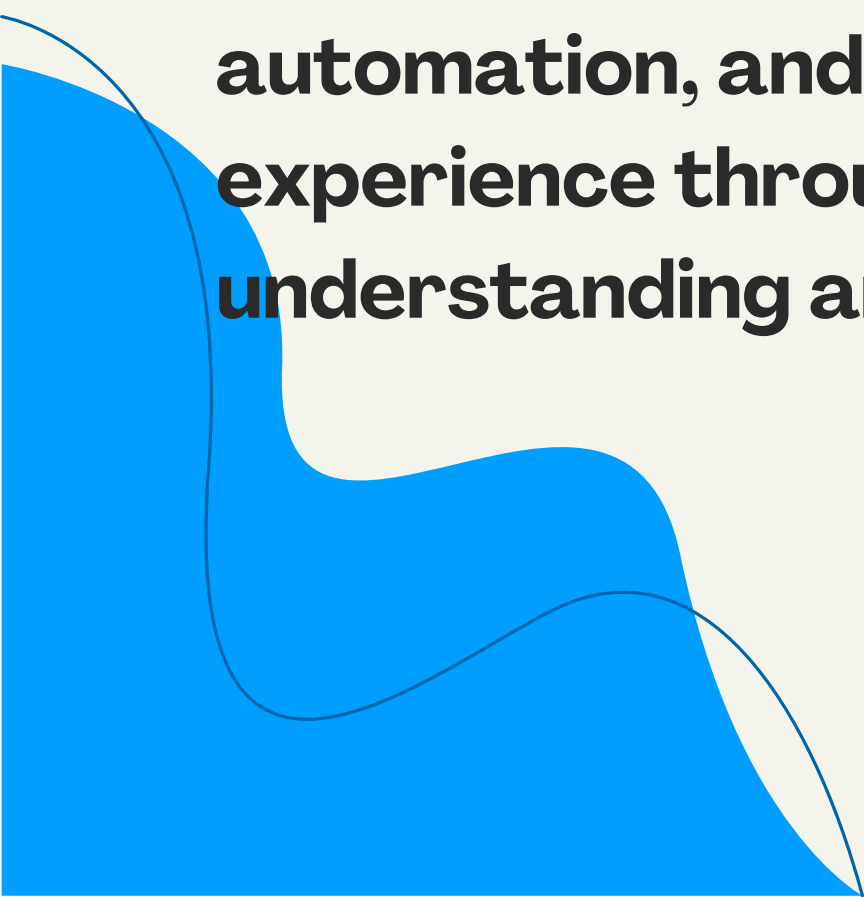
-Namsam Koyu Rai

-Shwopnil Nakarmi




# Chatbot

**A chatbot is an AI-powered tool designed to simulate human-like conversations, providing instant responses and assistance. It can be used for customer support, information retrieval, task automation, and enhancing user experience through natural language understanding and interaction.**



## **Purpose**

- **24/7 Availability**
  - **Cost-Effective**
  - **Enhanced User Experience**
  - **Scalability**
  - **Time-Saving**
  - **Lead Generation and Engagement**
  - **Multichannel Support**
  - **Data Collection and Insights**
- 

# Types of Chatbots

## • Retrieval-Based Chatbots

- **Description:** These chatbots rely on a predefined set of responses and select the most appropriate one based on user input.
- **Functionality:**
  - Use lookup tables or knowledge bases to provide answers.
  - Effective for straightforward queries where responses can be predetermined.
- **Use Cases:** Frequently Asked Questions (FAQs), basic customer support inquiries.

## • Generative Chatbots

- **Description:** These chatbots generate responses on the fly using machine learning models.
- **Functionality:**
  - Create unique responses based on the context of the conversation.
  - More complex and data-intensive, often using deep learning techniques.
- **Use Cases:** Conversational agents that require more nuanced interactions, such as virtual assistants.

# Chatbot (Seq2Seq)

## Step 1: Install and Import Libraries

```
# Step 1: Install required libraries
!pip install tensorflow pandas nltk

# Import necessary libraries
import os
import zipfile
import pandas as pd
import numpy as np
import tensorflow as tf
from tensorflow.keras.layers import Input, LSTM, Dense, Embedding, Attention, Concatenate
from tensorflow.keras.models import Model
import nltk
from nltk.tokenize import word_tokenize
from sklearn.model_selection import train_test_split

# Download necessary NLTK data
nltk.download('punkt')
nltk.download('punkt_tab')
```

# Libraries used

**NumPy**: For handling numerical operations and transformations efficiently

**Pandas**: To load and manipulate the dataset for training and testing

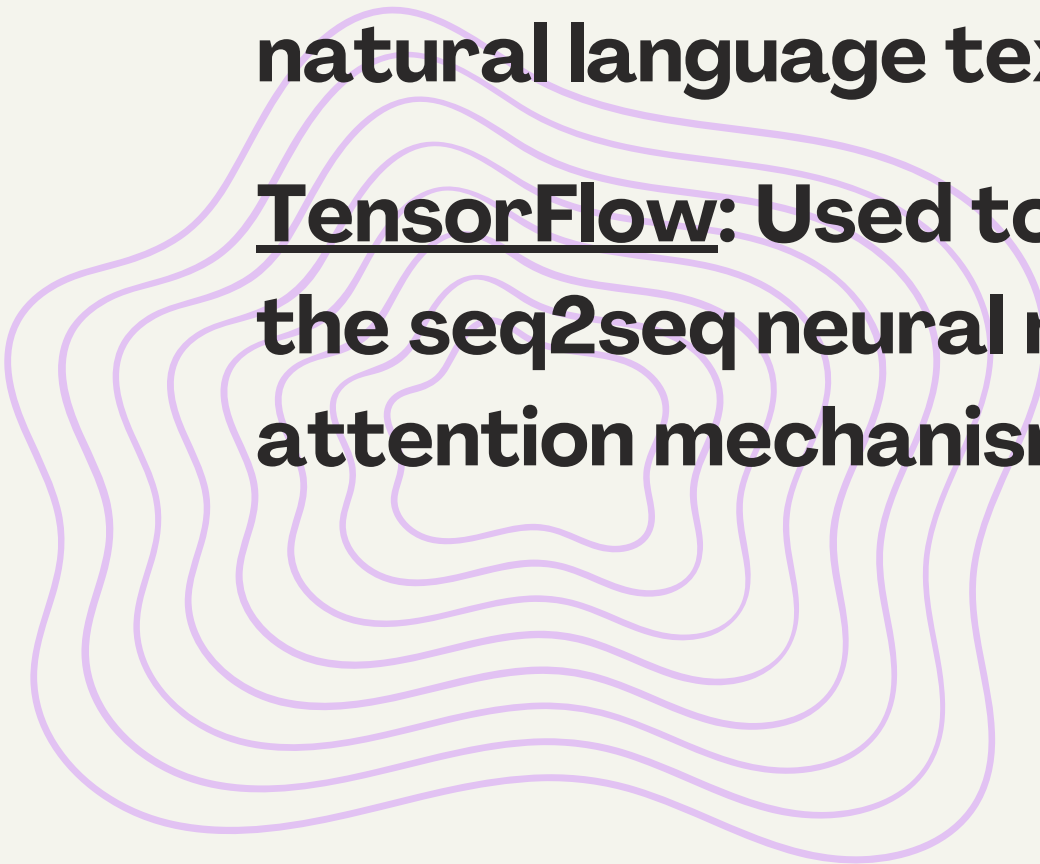
**NLTK** (**Natural Language Toolkit**): For natural language text processing

**TensorFlow**: Used to build and train the seq2seq neural network with attention mechanism.

**Scikit-learn**: For splitting the dataset into training and testing sets

**OS**: To interact with the operating system for file handling

**Zipfile**: To handle ZIP file extraction



# Step 2: Dataset Preparation

## Type of file

Text file with simple dialogues in two column where first one consists of questions and second column is answers.

```
hi, how are you doing?  i'm fine. how about yourself?  
i'm fine. how about yourself?  i'm pretty good. thanks for asking.  
i'm pretty good. thanks for asking.      no problem. so how have you been?  
no problem. so how have you been?      i've been great. what about you?  
i've been great. what about you?      i've been good. i'm in school right now.  
i've been good. i'm in school right now.      what school do you go to?  
what school do you go to?      i go to pcc.  
i go to pcc.      do you like it there?  
do you like it there?  it's okay. it's a really big campus.  
it's okay. it's a really big campus.      good luck with school.  
good luck with school.  thank you very much.  
how's it going? i'm doing well. how about you?  
i'm doing well. how about you?  never better, thanks.  
never better, thanks.      so how have you been lately?  
so how have you been lately?      i've actually been pretty good. you?  
i've actually been pretty good. you?      i'm actually in school right now.  
i'm actually in school right now.      which school do you attend?  
which school do you attend?      i'm attending pcc right now.  
i'm attending pcc right now.      are you enjoying it there?  
are you enjoying it there?      it's not bad. there are a lot of people there.  
it's not bad. there are a lot of people there.  good luck with that.  
good luck with that.      thanks.  
how are you doing today?      i'm doing great. what about you?
```

- **Extract Dataset**

- **The dataset is unzipped to ready for use**

```
with zipfile.ZipFile(uploaded_file_path, 'r') as zip_ref:  
    zip_ref.extractall(extracted_path)
```

- **Load the Dataset**

- **A tab-separated .txt file with two columns :input and response is assumed.**

```
df = pd.read_csv(dataset_file, delimiter='\t', header=None, names=['input', 'response'])
```



# Step 3: Data Preprocessing

- **Clean the Text:**

- **Convert to lowercase**

- **Example: “HELLO, How are You?” ==> “hello, how are you?”**

- **Tokenization**

- **Example: “hello, how are you?” ==> [ ‘hello’, ‘,’, ‘how’, ‘are’, ‘you’, ‘?’ ]**

- **Join tokens**

- **Example: [ ‘hello’, ‘,’, ‘how’, ‘are’, ‘you’, ‘?’ ] ==> “hello, how are you?”**

```
# Cleaning the text (lowercasing, tokenizing)
def preprocess_text(text):
    tokens = word_tokenize(str(text).lower())
    return ' '.join(tokens)
```




- **Apply Preprocessing**

```
df['input'] = df['input'].apply(preprocess_text)
df['response'] = df['response'].apply(preprocess_text)
```

- **Split the Dataset**

- **Splits the dataset into training (80%) and testing (20%) sets.**

```
# Splitting the dataset
train_data, test_data = train_test_split(df, test_size=0.2, random_state=42)
```



# Step 3: Tokenization and Padding

- **Fit the Tokenizer**

```
# Tokenization and vectorization
tokenizer = tf.keras.preprocessing.text.Tokenizer()
tokenizer.fit_on_texts(df['input'].tolist() + df['response'].tolist())
```

- **Convert to Sequences**

```
input_sequences = tokenizer.texts_to_sequences(train_data['input'])
response_sequences = tokenizer.texts_to_sequences(train_data['response'])
```



## • Pad Sequences

```
# Padding sequences
max_sequence_len = 50
input_sequences = tf.keras.preprocessing.sequence.pad_sequences(input_sequences, maxlen=max_sequence_len, padding='post')
response_sequences = tf.keras.preprocessing.sequence.pad_sequences(response_sequences, maxlen=max_sequence_len, padding='post')
```

