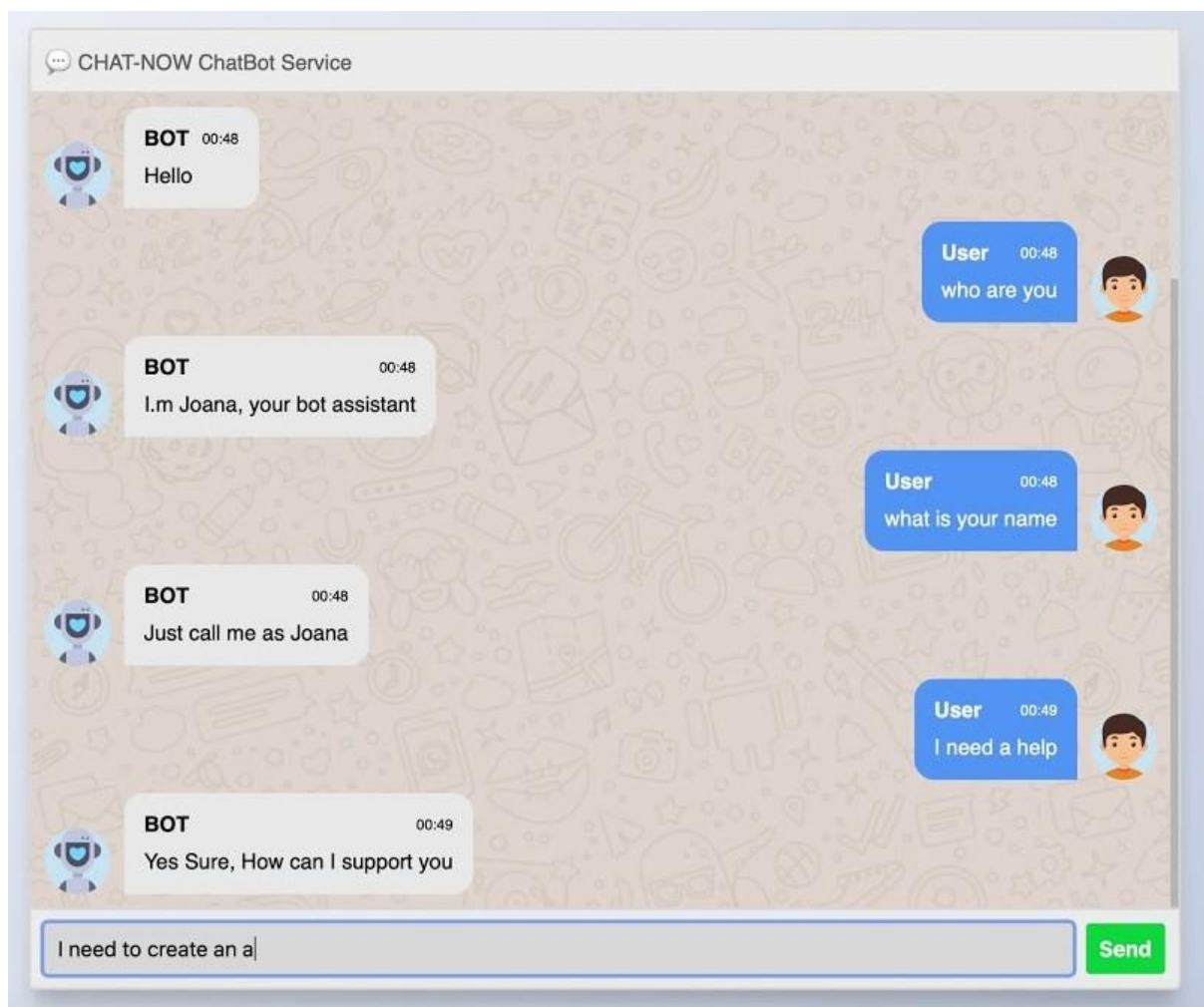


# CREATE A CHATBOT IN PYTHON

aut104302 - SIVAVETRIVEL. S

## ABSTRACT

This is an abstract about creating a chatbot in Python using data visualization, text cleaning, tokenization, encoder building, model training, metric visualization, and time to chat.



## DATA VISUALIZATION

Data visualization is the process of converting data into a graphical format that is easy to understand. This can be helpful for identifying patterns and trends in data, as well as for communicating data to others.

In the context of chatbot development, data visualization can be used to:

- Understand the distribution of user inputs and chatbot responses
- Identify the most common user queries
- Identify the most common chatbot errors
- Track the performance of the chatbot over time

## **TEXT CLEANING**

Text cleaning is the process of removing noise and inconsistencies from text data. This can include tasks such as removing punctuation, stop words, and slang. Text cleaning is important for chatbot development because it ensures that the chatbot is able to understand user input accurately.

## **TOKENIZATION**

Tokenization is the process of dividing text data into smaller units, such as words or characters. This is an important step in many natural language processing tasks, including chatbot development. Tokenization helps the chatbot to understand the meaning of user input and to generate appropriate responses.

## **ENCODER BUILDING**

An encoder is a neural network that is used to convert text data into a numerical representation. This representation is then used by the chatbot to generate responses. There are many different ways to build an encoder. One common approach is to use a recurrent neural network (RNN). RNNs are well-suited for encoding text data because they can learn long-term dependencies in the data.

## **MODEL TRAINING**

Once the encoder has been built, the chatbot model needs to be trained. This involves feeding the encoder examples of user inputs and chatbot responses. The model will learn to generate responses that are similar to the responses in the training data.

## **METRIC VISUALIZATION**

Once the model has been trained, it is important to visualize the metrics to assess its performance. This can include metrics such as accuracy, precision, and recall. Metric visualization can help to identify areas where the model needs to be improved.

## **TIME TO CHAT**

Once the model has been trained and evaluated, it is ready to be used to chat with users. The chatbot can be deployed on a variety of platforms, such as websites, mobile apps, and messaging platforms.

## **CONCLUSION**

Creating a chatbot in Python can be a complex task. However, by using data visualization, text cleaning, tokenization, encoder building, model training, metric visualization, and time to chat, it is possible to create a chatbot that is both accurate and engaging