

Chatbot for Mait Website Using AI machine learning (Building a neural network)

Team members.

1. Kishan Payadi (01314811621)
2. Sahil Verma (04014811621)

Professor.

- Ms. Gunjan Chugh(Assistant Professor-AI&ML)

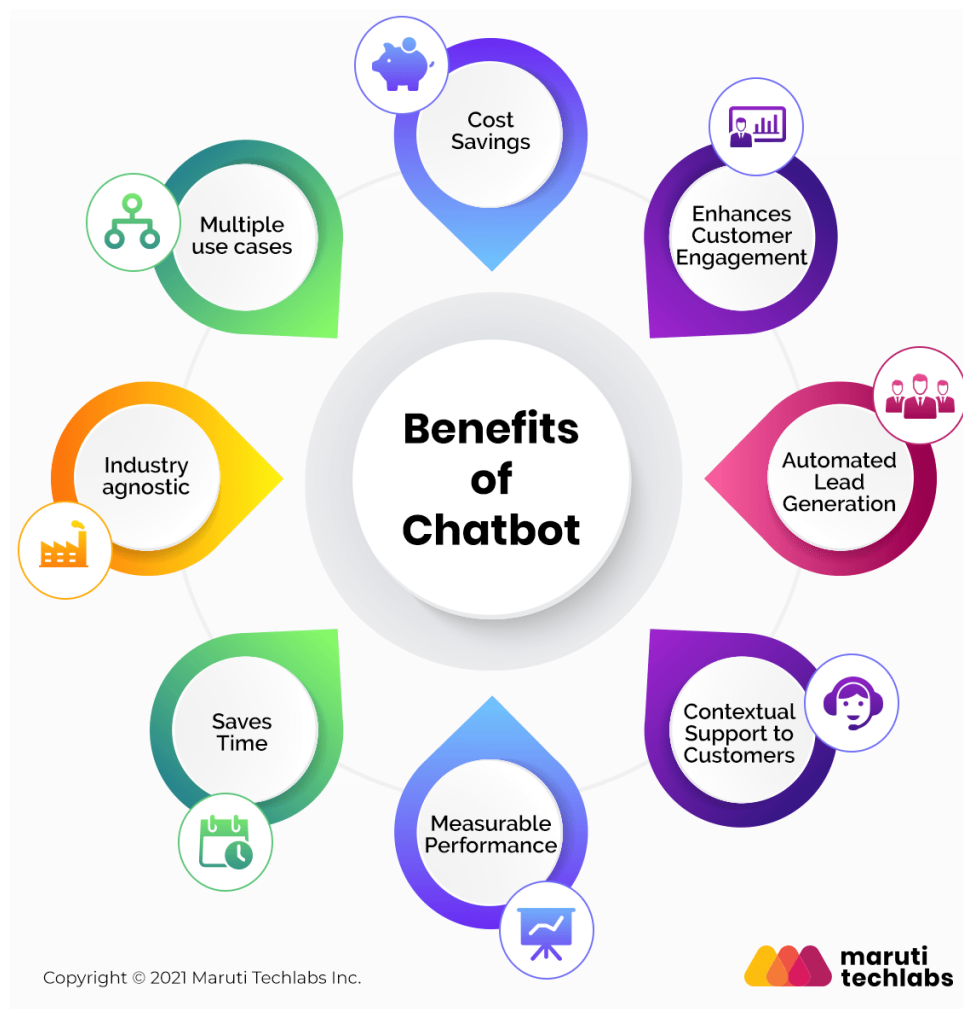


Abstract

What is chatbot

At the most basic level, a chatbot is a computer program that simulates and processes human conversation (either written or spoken), allowing humans to interact with digital devices as if they were communicating with a real person. Chatbots can be as simple as rudimentary programs that answer a simple query with a single-line response, or as sophisticated as digital assistants that learn and evolve to deliver increasing levels of personalization as they gather and process information.

Why should we care about a chat bot?



Introduction

Since we will be developing a Chatbot with Python using Machine Learning, we need some data to train our model. But we're not going to collect or download a large dataset we are gonna make our own data set because we are making a chatbot for our own college (MAIT) website.

To create this dataset to create a chatbot with Python, we need to understand what intents we are going to train. An "intention" is the user's intention to interact with a chatbot or the intention behind every message the chatbot receives from a particular user.

Therefore, it is vital to understand the good intentions of your chatbot depending on the domain you will be working with. So why does he need to define these intentions? This is an essential point to understand.

In order to answer questions asked by the users and perform various other tasks to continue conversations with the users, the chatbot really needs to understand what users are saying or have the intention to do. This is why your chatbot must understand the intentions behind users' messages.

How can you get your chatbot to understand the intentions so that users feel like they know what they want and provide accurate answers? The strategy here is to set different intents and create training samples for those intents and train your chatbot model with these sample training data as model training data (X) and intents as model training categories (Y).

How does our project work

In our project(Explain more)

1. We need to define a few simple intents and a group of messages that match those intents and also map some responses based on each intent category. I'll create a JSON file named "intents.json".
2. Now we create a chatbot with Python and Machine Learning is to prepare the data to train our chatbot. I'll start this step by importing the necessary libraries and packages.
3. Now I will read the JSON file and process the required files.
4. Now we need to use the label encoder method provided by the Scikit-Learn library in Python.
5. Now we need to vectorize the data using the Tokenization method to create a chatbot with Python and Machine Learning.
6. Now the next and most important step in the process of building a chatbot with Python and Machine Learning is to train a sequential neural network. Now, I will train and create a neural network to train our chatbot.
7. We've trained the model, but before we go any further in the process of building a chatbot with Python and Machine Learning, let's save the model so that we can use this neural network in the future as well.

8. Now I am going to implement a chat function to interact with a real user. When the message from the user will be received, the chatbot will compute the similarity between the sequence of the new text and the training data.
9. Taking into account the trust scores obtained for each category, it categorizes the user's message according to an intention with the highest trust score
10. And finally, one of the responses is chosen and then it is printed.

Technology used.

1. Tensor flow.
 - a. TensorFlow is a free and open-source software library for machine learning and artificial intelligence. It can be used across a range of tasks but has a particular focus on training and inference of deep neural networks.
2. Sklearn.
 - a. Scikit-learn (Sklearn) is the most useful and robust library for machine learning in Python. It provides a selection of efficient tools for machine learning and statistical modeling including classification, regression, clustering and dimensionality reduction via a consistency interface in Python. This library, which is largely written in Python, is built upon NumPy, SciPy and Matplotlib.
3. JSON
 - a. JSON stands for javascript object notation, it is the lightweight format for storing and transporting data.
4. Numpy
 - a. NumPy is a library for the Python programming language, adding support for large, multi-dimensional arrays and matrices, along with a large collection of high-level mathematical functions to operate on these arrays.
5. Keras
 - a. What is Keras and why it is used?
 - b. Keras is a high-level, deep-learning API developed by Google for implementing neural networks. It is written in Python and is used to make the implementation of neural networks easy.
 - c. Keras is an API designed for human beings, not machines. Keras follows best practices for reducing cognitive load
6. Tokenizer
 - a. Tokenization is used in natural language processing to split paragraphs and sentences into smaller units that can be more easily assigned meaning.
7. Pickle module.
 - a. Python pickle module is used for serializing and de-serializing a Python object structure. Any object in Python can be pickled so that it can be saved on disk. What pickle does is that it "serializes" the object first before writing it to file. Pickling is a way to convert a python object (list, dict, etc.)

Implementation of the code

```
3s ▶ import json
import numpy as np
import tensorflow as tf
from tensorflow import keras
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense, Embedding, GlobalAveragePooling1D
from tensorflow.keras.preprocessing.text import Tokenizer
from tensorflow.keras.preprocessing.sequence import pad_sequences
from sklearn.preprocessing import LabelEncoder
```

```
0s [2] with open('intents.json') as file:
      data = json.load(file)
```

```
0s [3] training_sentences = []
      training_labels = []
      labels = []
      responses = []
```

In this all the tags are included in the label and training sentence=pattern training label=tag responses=responses

```
0s [4] for intent in data['intents']:
      for pattern in intent['patterns']:
          training_sentences.append(pattern)
          training_labels.append(intent['tag'])
          responses.append(intent['responses'])

      if intent['tag'] not in labels:
          labels.append(intent['tag'])
```

Executing (9m 36s) Cell > chat() > raw_input() > _input_request() > select()

```
0s [5] num_classes = len(labels)
      print(training_labels)
```

```
['greeting', 'greeting', 'greeting', 'greeting', 'greeting', 'greeting', 'goodbye', 'goodbye', 'goodbye', 'thanks', 'thanks', 'thanks', 'th
```

```
0s [6] lbl_encoder = LabelEncoder()
      lbl_encoder.fit(training_labels)
      training_labels = lbl_encoder.transform(training_labels)
```

```
0s ▶ print(training_labels)
```

```
☐ [17 17 17 17 16 16 16 20 20 20 11 18 18 18 15 15 15 15 14 14 14 14
   14 14 3 3 3 7 1 13 0 10 19 2 5 6 12 9 8 4]
```

```
0s [8] print(training_sentences)
```

```
['Hi', 'Hey', 'Is anyone there?', 'Hello', 'Hay', 'Bye', 'See you later', 'Goodbye', 'Thanks', 'Thank you', "That's helpful", '
```

```
0s [9] vocab_size = 1000
      embedding_dim = 16
      max_len = 20
      oov_token = "<OOV>"

      tokenizer = Tokenizer(num_words=vocab_size, oov_token=oov_token)
      tokenizer.fit_on_texts(training_sentences)
      word_index = tokenizer.word_index
      sequences = tokenizer.texts_to_sequences(training_sentences)
      padded_sequences = pad_sequences(sequences, truncating='post', maxlen=max_len)
```

Executing (10m 17s) Cell > chat() > raw_input() > _input_request() > select()

```

✓ [10] print(training_sentences)
08 print(sequences)

['Hi', 'Hey', 'Is anyone there?', 'Hello', 'Hay', 'Bye', 'See you later', 'Goodbye', 'Thanks', 'Thank you', "That's helpful", 'Thanks for
[[23], [24], [7, 25, 26], [27], [28], [29], [30, 11, 31], [32], [15], [33, 11], [34, 35], [15, 36, 3, 37], [8, 5, 11], [6, 7, 16, 17], [6,

```

Need to learn about sequential model from scratch(Not much understood from here)

```

✓ [11] model = Sequential()
08 model.add(Embedding(vocab_size, embedding_dim, input_length=max_len))
model.add(GlobalAveragePooling1D())
model.add(Dense(16, activation='relu'))
model.add(Dense(16, activation='relu'))
model.add(Dense(num_classes, activation='softmax'))

```

```

✓ [12] print(model)
08 <keras.engine.sequential.Sequential object at 0x7f5e3aca1d60>

```

```

✓ [13] model.compile(loss='sparse_categorical_crossentropy',
08 optimizer='adam', metrics=['accuracy'])

```

```

✓ [14] model.summary()
08 epochs = 500
history = model.fit(padded_sequences, np.array(training_labels), epochs=epochs)

```

```

✓ [15] # to save the trained model
08 model.save("chat_model")

```

```

✓ [16] import pickle
08
# to save the fitted tokenizer
with open('tokenizer.pickle', 'wb') as handle:
    pickle.dump(tokenizer, handle, protocol=pickle.HIGHEST_PROTOCOL)

# to save the fitted label encoder
with open('label_encoder.pickle', 'wb') as ecn_file:
    pickle.dump(lbl_encoder, ecn_file, protocol=pickle.HIGHEST_PROTOCOL)

```

```

✓ [16]
08

```

```

def chat():
    # load trained model
    model = keras.models.load_model('chat_model')

    # load tokenizer object
    with open('tokenizer.pickle', 'rb') as handle:
        tokenizer = pickle.load(handle)

    # load label encoder object
    with open('label_encoder.pickle', 'rb') as enc:
        lbl_encoder = pickle.load(enc)

```

```
# load label encoder object
with open('label_encoder.pickle', 'rb') as enc:
    lbl_encoder = pickle.load(enc)

# parameters
max_len = 20

> while True:
    print("User: ")
    inp = input()
    if inp.lower() == "quit":
        break

    result = model.predict(keras.preprocessing.sequence.pad_sequences(tokenizer.texts_to_sequences([inp]),
                                                                    truncating='post', maxlen=max_len))
    tag = lbl_encoder.inverse_transform([np.argmax(result)])

    for i in data['intents']:
        if i['tag'] == tag:
            print("ChatBot:" + np.random.choice(i['responses']))

print("Start messaging with the bot (type quit to stop)!")
chat()
```

Screenshot of the implementation of the code

```
Start messaging with the bot (type quit to stop):
User:
Hi
... 1/1 [=====] - 0s 95ms/step
ChatBot:Hi
User:
Who are you?
1/1 [=====] - 0s 21ms/step
ChatBot:I'am your bot assistant here to help you with MAIT website. Visit here for any issues https://mait.ac.in/
User:
what should I call you
1/1 [=====] - 0s 25ms/step
ChatBot:You can call me MAIT_BUDDY
User:
How beautiful is Mait campus
1/1 [=====] - 0s 17ms/step
ChatBot:http://gallery.mait.ac.in/gallery3/ follow here for photo galary of mait
User:
How beautiful is Mait campus
1/1 [=====] - 0s 18ms/step
ChatBot:http://gallery.mait.ac.in/gallery3/ follow here for photo galary of mait
User:
Do we get placement after degree
1/1 [=====] - 0s 17ms/step
ChatBot:https://mait.ac.in/index.php/placements/about-placements follow here for placement details of MAIT
User:

Is there any fees wavier in mait
1/1 [=====] - 0s 20ms/step
ChatBot:https://scholarships.gov.in/ follow this for scholarship or https://www.vidyalakshmi.co.in/Students/ follow this for education loan
User:
Is there any fees wavier in mait
1/1 [=====] - 0s 21ms/step
ChatBot:https://scholarships.gov.in/ follow this for scholarship or https://www.vidyalakshmi.co.in/Students/ follow this for education loan
User:

What are some extra cocircular activities of mait
1/1 [=====] - 0s 18ms/step
ChatBot:https://mait.ac.in/index.php/societies/technical Visit here for Technical socities
User:
what is taught in MAIT
1/1 [=====] - 0s 25ms/step
ChatBot:https://mait.ac.in/index.php/academics/syllabus
User:
What all courses does MAIT provides
1/1 [=====] - 0s 19ms/step
ChatBot:CSE, ECE, EEE, IT, MAE, ME, CST, ITE, AIML, AIDS
User:
Seniors are troubeling me
1/1 [=====] - 0s 22ms/step
ChatBot:https://mait.ac.in/images/Anti-Ragging-Measures-2022.pdf
User:
Thanks for the help
1/1 [=====] - 0s 18ms/step
ChatBot:Happy to help!
User:

```


References

All the links to the website of tutorials of TensorFlow etc.

1. Mait official website =><https://mait.ac.in/>
2. Tensorflow. =><https://www.tensorflow.org/>
3. Sklearn tutorial =><https://www.tensorflow.org/>
4. JSON =><https://www.json.org/json-en.html>
5. Numpy tutorial=><https://numpy.org/>
6. Sklearn =><https://numpy.org/>
7. sklearn.preprocessing.LabelEncoder =><https://numpy.org/>
8. Tokenizers =><https://docs.python.org/3/library/tokenize.html>
9. Tensorflow =>https://www.tensorflow.org/guide/keras/sequential_model
10. Pickle module =><https://docs.python.org/3/library/pickle.html#:~:text=The%20pickle%20module%20can%20transform,with%20the%20same%20internal%20structure.>