July 2021

FAQ CHATBOT

Prepared by: Srivathsan

About

An Covid19 FAQ (Frequently Asked Questions) ChatBot is a type of internet bot or software application that is beneficial for answering some of the most frequently asked questions related to covid19. This FAQ bots helps the customer's be aware about the pandemic.

Mission and Vision

Never in the history of public health, a pandemic of a disease threatened the humanity as COVID-19, technical name of a newly identified coronavirus, has inflicted. The disease is caused by the SARS-coronavirus-2, a virus primarily zoonotic and was not found in humans. WHO has declared COVID-19 a global pandemic and a public health emergency. The spread of corona epidemic is unprecedented and has reached 199 countries and territories around the world (and the cruise ship Diamond Princess harbored in Yokohama, Japan), and has affected over 556,141 people, testing positive for coronavirus. The death toll has reached 25,237 (Worldometer, March 27, 2020; 14.36 GMT). Pandemic has spread with a high velocity across the globe within a short period of time. Unlike SARS (2003) and MERS (2012), the case fatality rate is higher at 2–3%.Inorder to educate the public about this serious threat, this chatbot is manufactured to create an awareness amongst the public.

Concept

Since we are going to develop a deep learning based model, we need data to train our model. So we can just create our own dataset in order to train the model. So an intent is created in a JSON file format. The concept here is to define different intents and make the training sample for those intents and train the chatbot model with those training sample data as model training data(X) and intents as model training category(Y)

Implementation

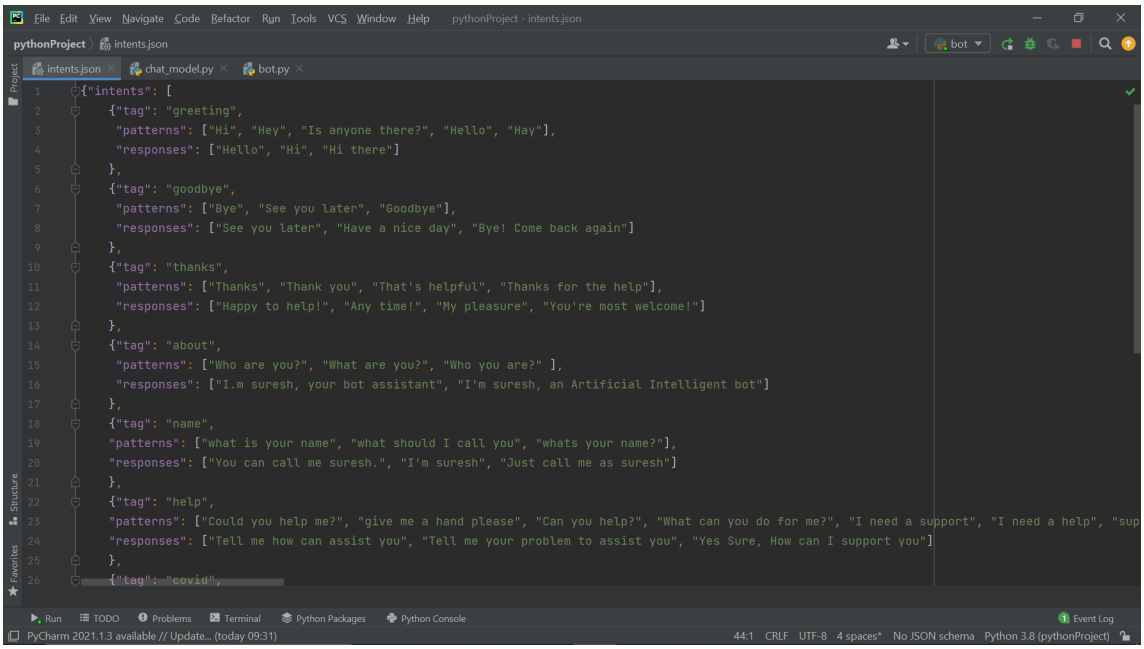
The packages that are used are:

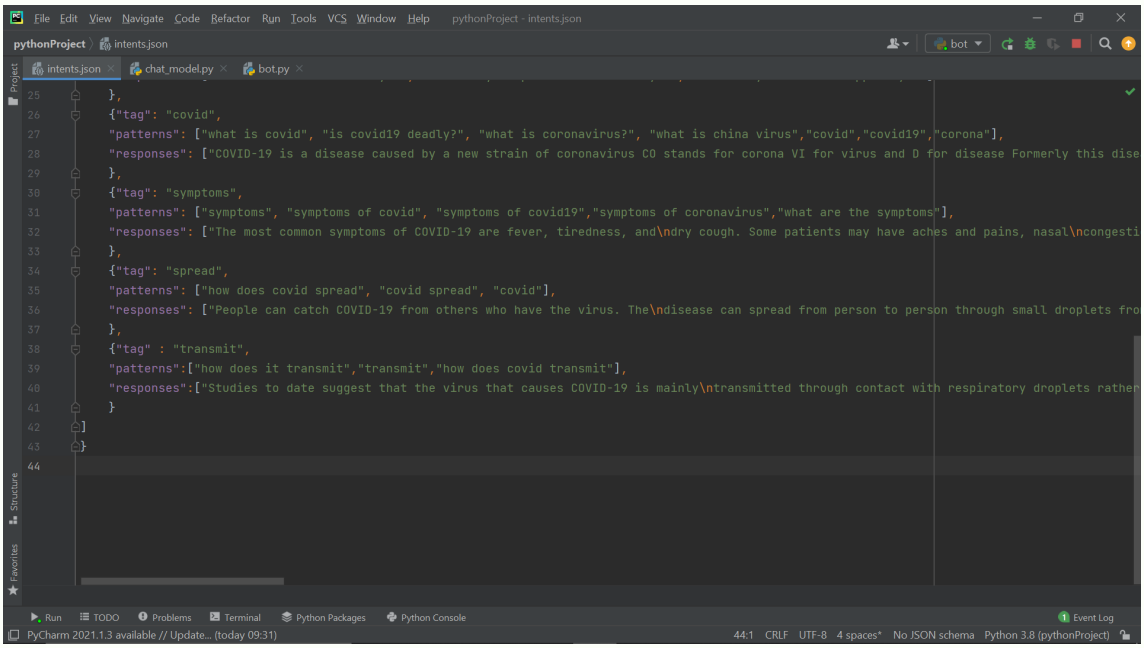
- Tensorflow
- NLTK
- Colorama
- Numpy
- Scikit-Learn

02

Intents

Few simple intents were defined and bunch of messages that corresponds to those intents and also mapped some responses according to each intent category.Created a JSON file named “intents.json” including these data

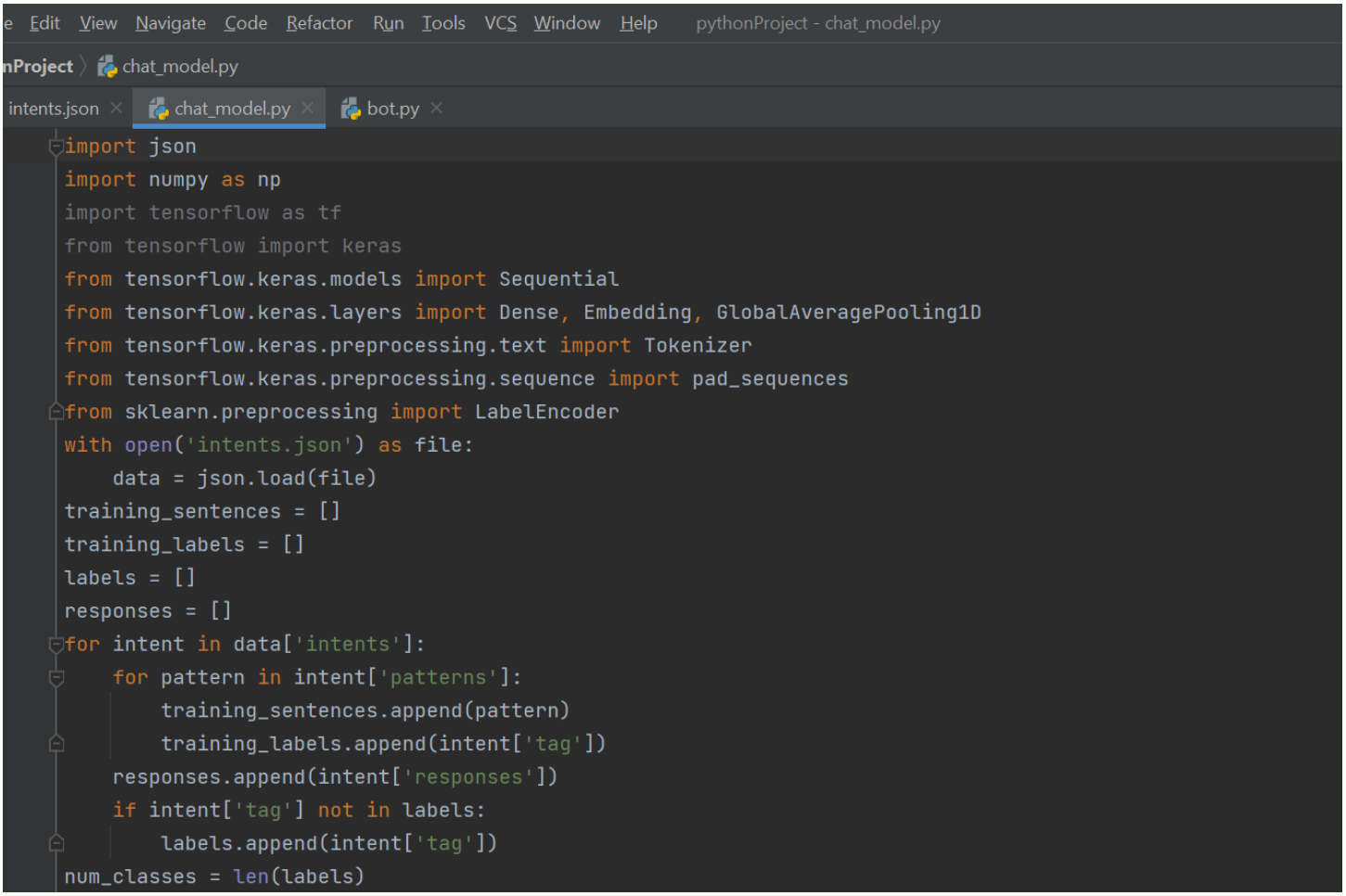




02

Preparing the data

After importing all the required packages,We have to load the JSON file and extract the data



The variable "*training_sentences*" holds all the training data (which are the sample messages in each intent) and the "*training_labels*" variable holds all the target labels correspond to each training data. Then we use "LabelEncoder()" function is used which is provided by scikit-learn to convert the target labels into a model understandable form.

```
lbl_encoder = LabelEncoder()
lbl_encoder.fit(training_labels)
training_labels = lbl_encoder.transform(training_labels)
```

Now, to the part of vectorizing our text data corpus by using the "*Tokenizer*" class. This will allows to limit the vocabulary size up to some defined number.
When using this class for the text pre-processing task, by default all punctuations will be removed, turning the texts into space-separated sequences of words, and these sequences are then split into lists of tokens, which will be indexed or vectorized.
Adding the "oov_token" which is a value for "out of token" to manage the vocabulary words(tokens) at inference time.

02

The variable "*pad_sequences*" is used to make all the training text sequences into the same size.

```
vocab_size = 1000
embedding_dim = 16
max_len = 20
oov_token = "<OOV>"
tokenizer = Tokenizer(num_words=vocab_size, oov_token=oov_token) # adding out of vocabulary 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)
```

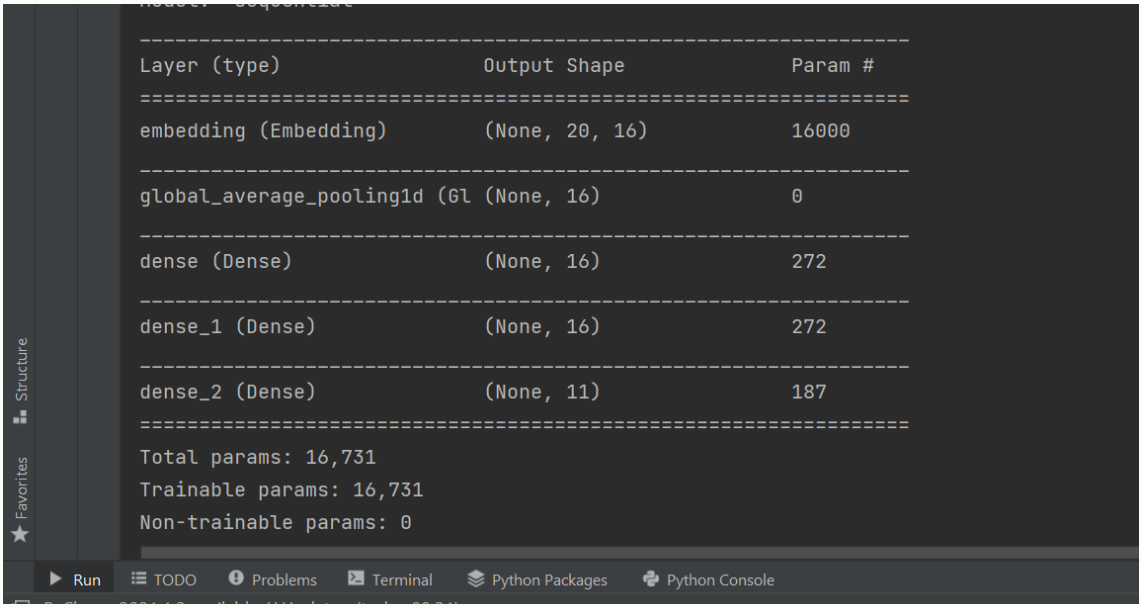
Training the model

A neural network archietecture is defined for the model.
A seqential model class of keras is used

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

02

This is our model's archietecture :



Model is fitted using the training data's and labels

```
epochs = 550
history = model.fit(padded_sequences, np.array(training_labels), epochs=epochs)
```

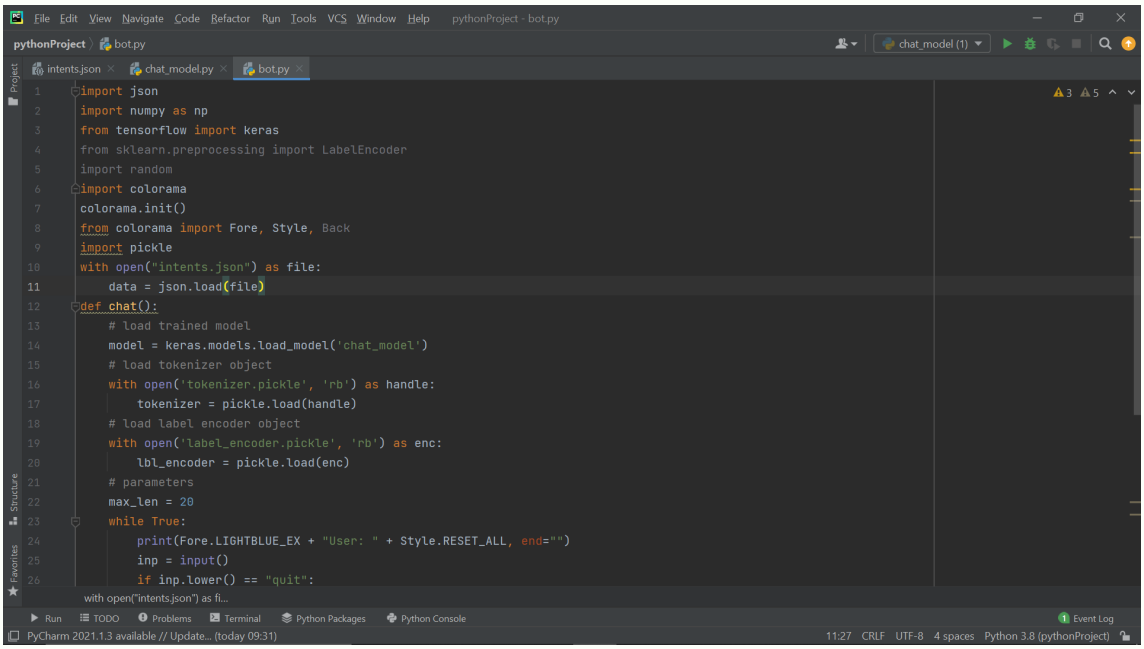
Saving the model

```
model.save("chat_model")
import pickle
# saving tokenizer
with open('tokenizer.pickle', 'wb') as handle:
    pickle.dump(tokenizer, handle, protocol=pickle.HIGHEST_PROTOCOL)
# saving label encoder
with open('label_encoder.pickle', 'wb') as ecn_file:
    pickle.dump(lbl_encoder, ecn_file, protocol=pickle.HIGHEST_PROTOCOL)
```

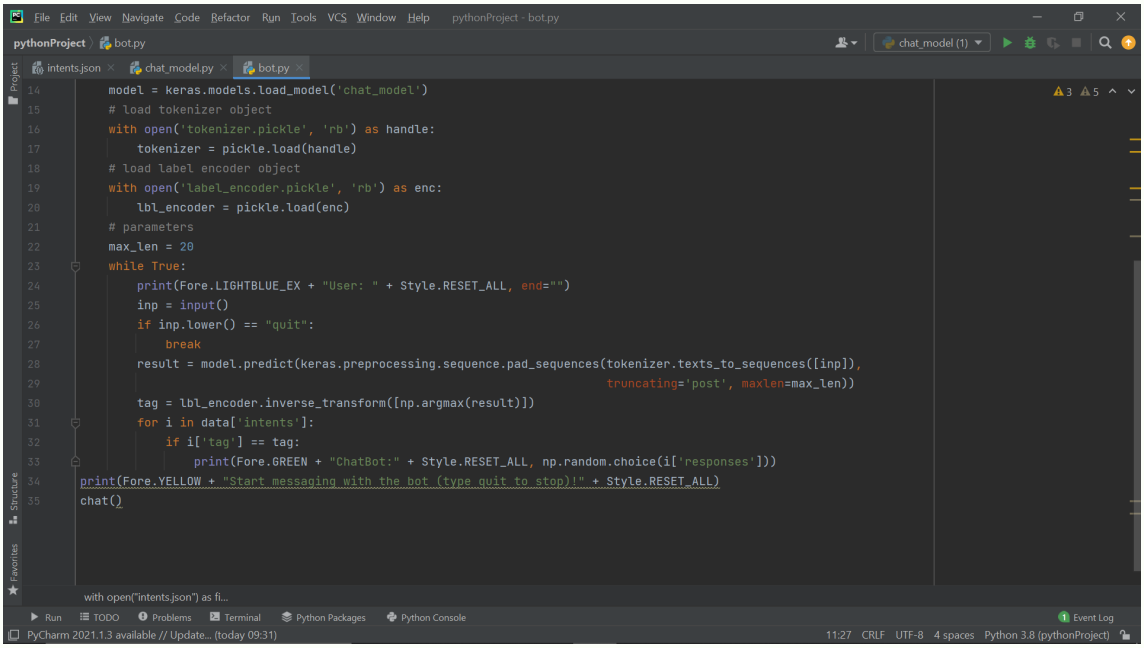
02

Implementing a chat function

A chat function is created in which when a messaged is recieved from the user the chat bot will calculate the similarity between the test sequence and training data

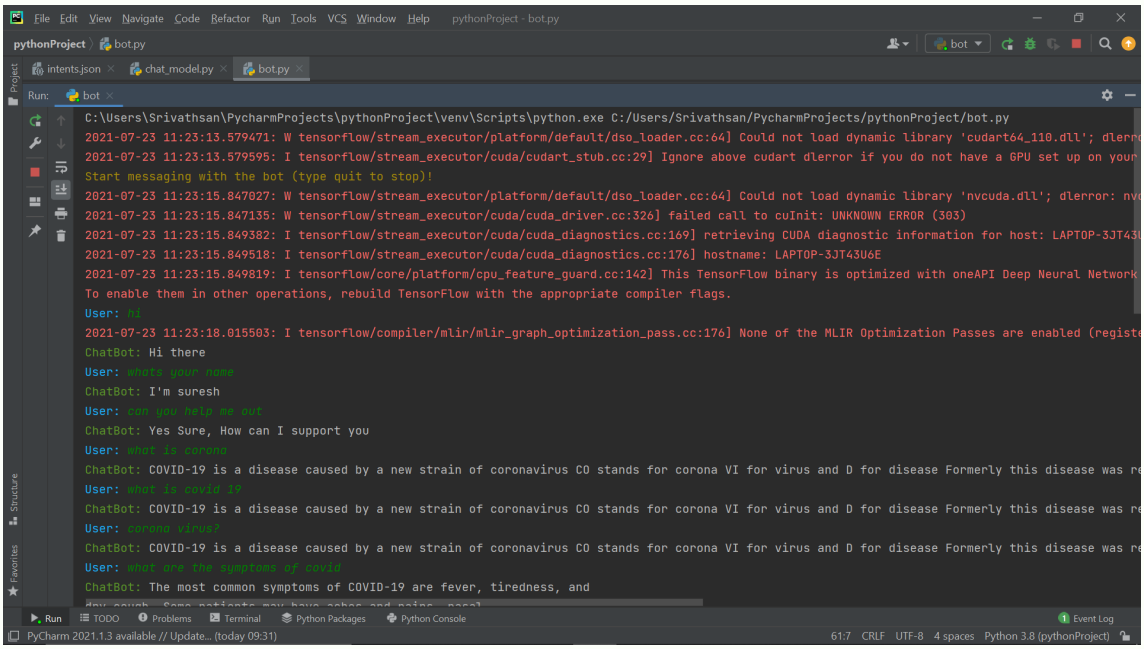


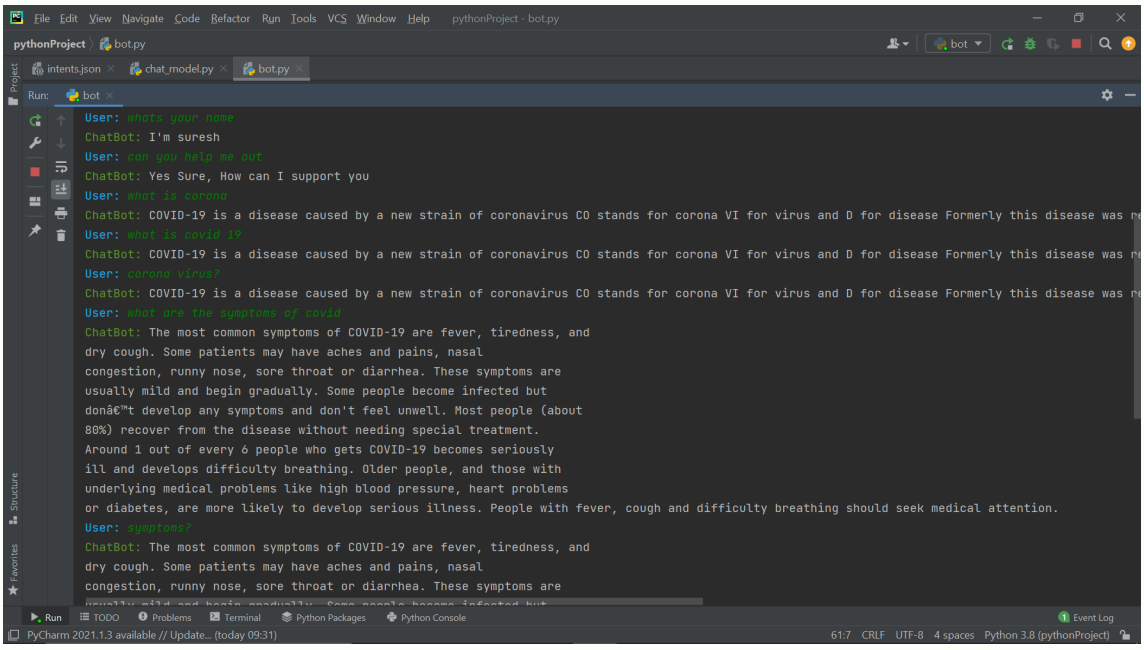
```
import json
import numpy as np
from tensorflow import keras
from sklearn.preprocessing import LabelEncoder
import random
import colorama
colorama.init()
from colorama import Fore, Style, Back
import pickle
with open("intents.json") as file:
    data = json.load(file)
def chat():
    # load trained 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)
    # parameters
    max_len = 20
    while True:
        print(Fore.LIGHTBLUE_EX + "User: " + Style.RESET_ALL, end="")
        inp = input()
        if inp.lower() == "quit":
```

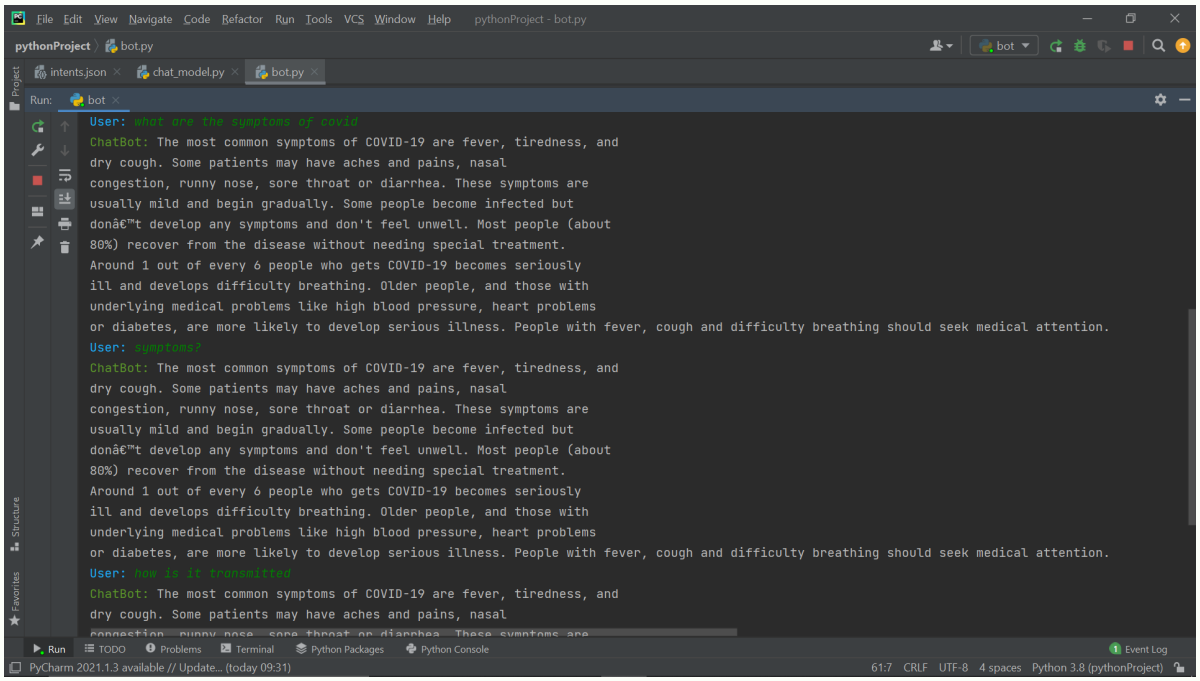


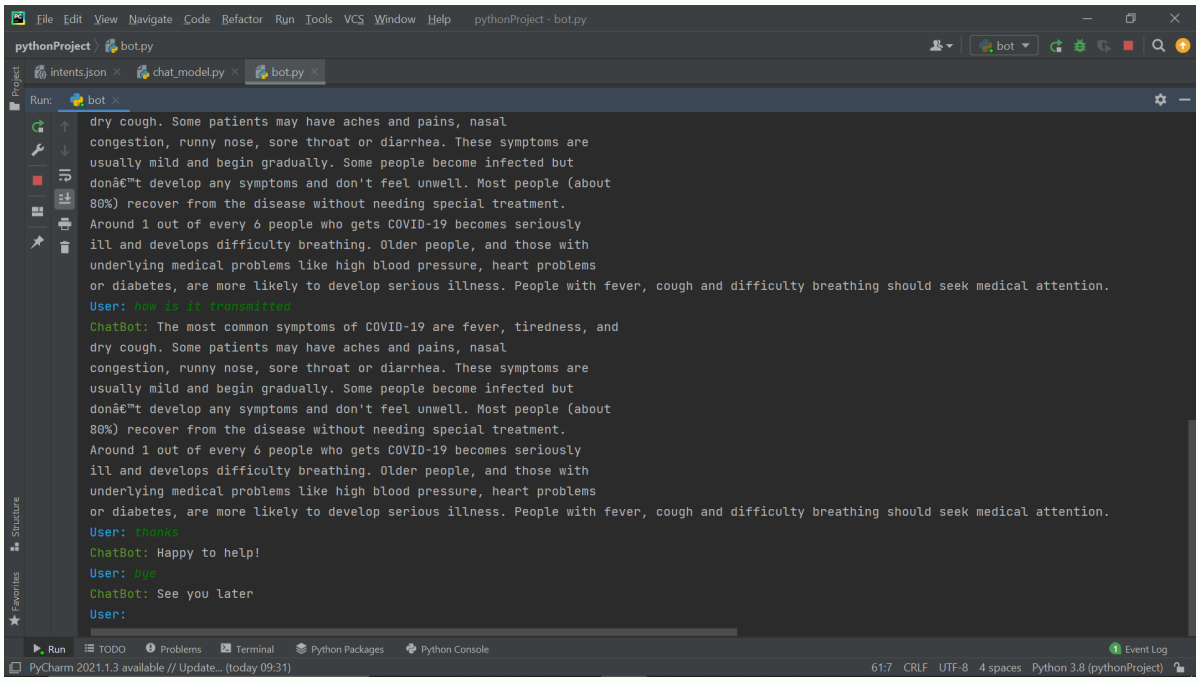
```
    model = keras.models.load_model('chat_model')
    # load tokenizer object
    with open('tokenizer.pickle', 'rb') as handle:
        tokenizer = pickle.load(handle)
    # load label encoder object
    with open('label_encoder.pickle', 'rb') as enc:
        lbl_encoder = pickle.load(enc)
    # parameters
    max_len = 20
    while True:
        print(Fore.LIGHTBLUE_EX + "User: " + Style.RESET_ALL, end="")
        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(Fore.GREEN + "ChatBot:" + Style.RESET_ALL, np.random.choice(i['responses']))
print(Fore.YELLOW + "Start messaging with the bot (type quit to stop)!" + Style.RESET_ALL)
chat()
```

Output


File Edit View Navigate Code Refactor Run Tools VCS Window Help pythonProject - bot.py
pythonProject bot.py
intents.json chat_model.py bot.py
Run: bot
C:\Users\Srivathsan\PycharmProjects\pythonProject\venv\Scripts\python.exe C:/Users/Srivathsan/PycharmProjects/pythonProject/bot.py
2021-07-23 11:23:13.579471: W tensorflow/stream_executor/platform/default/dso_loader.cc:64] Could not load dynamic library 'cudart64_110.dll'; dlerr
2021-07-23 11:23:13.579595: I tensorflow/stream_executor/cuda/cudart_stub.cc:29] Ignore above cudart dlerror if you do not have a GPU set up on your
Start messaging with the bot (type quit to stop)!
2021-07-23 11:23:15.847027: W tensorflow/stream_executor/platform/default/dso_loader.cc:64] Could not load dynamic library 'nvcuda.dll'; dlerror: nv
2021-07-23 11:23:15.847135: W tensorflow/stream_executor/cuda/cuda_driver.cc:326] failed call to cuInit: UNKNOWN ERROR (303)
2021-07-23 11:23:15.849382: I tensorflow/stream_executor/cuda/cuda_diagnostics.cc:169] retrieving CUDA diagnostic information for host: LAPTOP-3JT43
2021-07-23 11:23:15.849518: I tensorflow/stream_executor/cuda/cuda_diagnostics.cc:176] hostname: LAPTOP-3JT43U6E
2021-07-23 11:23:15.849819: I tensorflow/core/platform/cpu_feature_guard.cc:142] This TensorFlow binary is optimized with oneAPI Deep Neural Network
To enable them in other operations, rebuild TensorFlow with the appropriate compiler flags.
User: hi
2021-07-23 11:23:18.015503: I tensorflow/compiler/mlir/mlir_graph_optimization_pass.cc:176] None of the MLIR Optimization Passes are enabled (regist
ChatBot: Hi there
User: whats your name
ChatBot: I'm suresh
User: can you help me out
ChatBot: Yes Sure, How can I support you
User: what is corona
ChatBot: COVID-19 is a disease caused by a new strain of coronavirus CO stands for corona VI for virus and D for disease Formerly this disease was re
User: what is covid 19
ChatBot: COVID-19 is a disease caused by a new strain of coronavirus CO stands for corona VI for virus and D for disease Formerly this disease was re
User: corona virus?
ChatBot: COVID-19 is a disease caused by a new strain of coronavirus CO stands for corona VI for virus and D for disease Formerly this disease was re
User: what are the symptoms of covid
ChatBot: The most common symptoms of COVID-19 are fever, tiredness, and
Run TODO Problems Terminal Python Packages Python Console Event Log
PyCharm 2021.1.3 available // Update... (today 09:31) 61:7 CRLF UTF-8 4 spaces Python 3.8 (pythonProject)


File Edit View Navigate Code Refactor Run Tools VCS Window Help pythonProject - bot.py
pythonProject bot.py
intents.json chat_model.py bot.py
Run: bot
User: whats your name
ChatBot: I'm suresh
User: can you help me out
ChatBot: Yes Sure, How can I support you
User: what is corona
ChatBot: COVID-19 is a disease caused by a new strain of coronavirus CO stands for corona VI for virus and D for disease Formerly this disease was re
User: what is covid 19
ChatBot: COVID-19 is a disease caused by a new strain of coronavirus CO stands for corona VI for virus and D for disease Formerly this disease was re
User: corona virus?
ChatBot: COVID-19 is a disease caused by a new strain of coronavirus CO stands for corona VI for virus and D for disease Formerly this disease was re
User: what are the symptoms of covid
ChatBot: The most common symptoms of COVID-19 are fever, tiredness, and
dry cough. Some patients may have aches and pains, nasal
congestion, runny nose, sore throat or diarrhea. These symptoms are
usually mild and begin gradually. Some people become infected but
donâ€™t develop any symptoms and don't feel unwell. Most people (about
80%) recover from the disease without needing special treatment.
Around 1 out of every 6 people who gets COVID-19 becomes seriously
ill and develops difficulty breathing. Older people, and those with
underlying medical problems like high blood pressure, heart problems
or diabetes, are more likely to develop serious illness. People with fever, cough and difficulty breathing should seek medical attention.
User: symptoms?
ChatBot: The most common symptoms of COVID-19 are fever, tiredness, and
dry cough. Some patients may have aches and pains, nasal
congestion, runny nose, sore throat or diarrhea. These symptoms are
Run TODO Problems Terminal Python Packages Python Console Event Log
PyCharm 2021.1.3 available // Update... (today 09:31) 61:7 CRLF UTF-8 4 spaces Python 3.8 (pythonProject)





Thus you can see our chatbot working !!