

Python Chatbot Project – Learn to build your first chatbot using NLTK & Keras

How amazing it is to tell someone everything and anything and not being judged at all. A top class feeling it is and that's what the beauty of a chatbot is.

This is the 9th project in the 20 Python projects series by DataFlair and make sure to bookmark other interesting projects:

What is Chatbot?

A chatbot is an intelligent piece of software that is capable of communicating and performing actions similar to a human. Chatbots are used a lot in customer interaction, marketing on social network sites and instantly messaging the client. There are two basic types of chatbot models based on how they are built; Retrieval based and Generative based models.

1. Retrieval based Chatbots

A retrieval-based chatbot uses predefined input patterns and responses. It then uses some type of heuristic approach to select the appropriate response. It is widely used in the industry to make goal-oriented chatbots where we can customize the tone and flow of the chatbot to drive our customers with the best experience.

2. Generative based Chatbots

Generative models are not based on some predefined responses.

They are based on seq 2 seq neural networks. It is the same idea as machine translation. In machine translation, we translate the source code from one language to another language but here, we are going to transform input into an output. It needs a large amount of data and it is based on Deep Neural networks.

About the Python Project – Chatbot

In this Python project with source code, we are going to build a chatbot using deep learning techniques. The chatbot will be trained on the dataset which contains categories (intents), pattern and responses. We use a special recurrent neural network (LSTM) to classify which category the user's

message belongs to and then we will give a random response from the list of responses.

Let's create a retrieval based chatbot using NLTK, Keras, Python, etc.

Download Chatbot Code & Dataset

The dataset we will be using is 'intents.json'. This is a JSON file that contains the patterns we need to find and the responses we want to return to the user.

Please download python chatbot code & dataset from the following link:

[Python Chatbot Code & Dataset](#)

Prerequisites

The project requires you to have good knowledge of Python, Keras, and [Natural language processing \(NLTK\)](#). Along with them, we will use some helping modules which you can download using the python-pip command.

pip install tensorflow, keras, pickle, nltk

How to Make Chatbot in Python?

Now we are going to build the chatbot using Python but first, let us see the file structure and the type of files we will be creating:

- **Intents.json** – The data file which has predefined patterns and responses.
- **train_chatbot.py** – In this Python file, we wrote a script to build the model and train our chatbot.
- **Words.pkl** – This is a pickle file in which we store the words Python object that contains a list of our vocabulary.
- **Classes.pkl** – The classes pickle file contains the list of categories.
- **Chatbot_model.h5** – This is the trained model that contains information about the model and has weights of the neurons.
- **Chatgui.py** – This is the Python script in which we implemented GUI for our chatbot. Users can easily interact with the bot.

Here are the 5 steps to create a chatbot in Python from scratch:

1. Import and load the data file
2. Preprocess data
3. Create training and testing data
4. Build the model
5. Predict the response

6. 1. Import and load the data file

7. First, make a file name as train_chatbot.py. We import the necessary packages for our chatbot and initialize the variables we will use in our Python project.
8. The data file is in JSON format so we used the json package to parse the JSON file into Python.

2. Preprocess data

When working with text data, we need to perform various preprocessing on the data before we make a machine learning or a deep learning model. Based on the requirements we need to apply various operations to preprocess the data.

Tokenizing is the most basic and first thing you can do on text data. Tokenizing is the process of breaking the whole text into small parts like words.

Here we iterate through the patterns and tokenize the sentence using nltk.word_tokenize() function and append each word in the words list. We also create a list of classes for our tags.

Now we will lemmatize each word and remove duplicate words from the list. Lemmatizing is the process of converting a word into its lemma form and then creating a pickle file to store the Python objects which we will use while predicting.

3. Create training and testing data

4. Build the model

We have our training data ready, now we will build a deep neural network that has 3 layers. We use the Keras sequential API for this. After training the model for 200 epochs, we achieved 100% accuracy on our model. Let us save the model as 'chatbot_model.h5'.

5. Predict the response (Graphical User Interface)

To predict the sentences and get a response from the user to let us create a new file 'chatapp.py'.

We will load the trained model and then use a graphical user interface that will predict the response from the bot. The model will only tell us the class it belongs to, so we will implement some functions which will identify the class and then retrieve us a random response from the list of responses.

Again we import the necessary packages and load the 'words.pkl' and 'classes.pkl' pickle files which we have created when we trained our model:

To predict the class, we will need to provide input in the same way as we did while training. So we will create some functions that will perform text preprocessing and then predict the class.

After predicting the class, we will get a random response from the list of intents.

Now we will develop a graphical user interface. Let's use Tkinter library which is shipped with tons of useful libraries for GUI. We will take the input message from the user and then use the helper functions we have created to get the response from the bot and display it on the GUI. Here is the full source code for the GUI.

6. Run the chatbot

To run the chatbot, we have two main files; **train_chatbot.py** and **chatapp.py**. First, we train the model using the command in the terminal:

```
python train_chatbot.py
```

If we don't see any error during training, we have successfully created the model. Then to run the app, we run the second file.

```
python chatgui.py
```

The program will open up a GUI window within a few seconds. With the GUI you can easily chat with the bot.

Summary

In this Python data science project, we understood about chatbots and implemented a deep learning version of a chatbot in Python which is accurate. You can customize the data according to business requirements and train the chatbot with great accuracy. Chatbots are used everywhere and all businesses are looking forward to implementing bot in their workflow.

I hope you will practice by customizing your own chatbot using Python and don't forget to show us your work. And, if you found the article useful, do share the project with your friends and colleagues.