

Step one: Importing libraries

Imports are critical for successfully organizing your Python code. Correctly importing code will increase your productivity by allowing you to reuse code while also maintaining the maintainability of your projects.

The necessary libraries include:

JSON: It is possible to utilize it to work with JSON data.

String: Provides access to several potentially valuable constants.

Random: For various distributions, this module implements pseudo-random number generators.

WordNetLemmatizer: It can lemmatize.

Tensorflow: A multidimensional array of elements is represented by this symbol.

Sequential: Sequential groups a linear stack of layers into a `tf.keras.Model`.

Step two: Creating a JSON file

This step will create an intents JSON file that lists all the possible outcomes of user interactions with our chatbot. We first need a set of tags that users can use to categorize their queries. These tags include name, age, and many others. Every new tag would require a unique pattern.

Identifying these trends can help the chatbot train itself on how people query about our chatbot's name, allowing it to be more responsive. The chatbot will return pre-programmed responses to answer questions:

Step three: Processing data

In this section, vocabulary of all the terms used in the patterns, list of tag classes, list of all the patterns in the intents file, and all the related tags for each pattern will be created before creating our training data:

Step four: Designing a neural network model

Because neural networks can only understand numerical values, we must first process our data so that a neural network can understand what we are doing.

Step five: Building useful features

In order to make use of our model in a chatbot, we must first implement the necessary functionality, which will be made easier by building a library of utility functions will help: