



Build a Chatbot with Python

Deepika Singh

Apr 13, 2020 • 7 Min read • 456 Views

Apr 13, 2020 • 7 Min read • 456 Views

Data

Python

- [Introduction](#)
- [Baseline Libraries](#)
- [Building the Chatbot](#)
- [Conclusion](#)
- [Top ^](#)

Introduction

Conversational *NLP*, or natural language processing, is playing a big part in text analytics through chatbots. A chatbot is an artificial intelligence based tool built to converse with humans in their native language. These chatbots have become popular across industries, and are considered one of the most useful applications of natural language processing.

In this guide, you will learn to build your first chatbot using Python.

Baseline Libraries

You'll be using the `nltk` library in this guide. NLTK stands for *Natural Language Toolkit* and is a leading python library to work with text data. The first line of code below imports the library,



- [Introduction](#)
- [Baseline Libraries](#)
- [Building the Chatbot](#)
- [Conclusion](#)
- [Top ^](#)

while the second line uses the `nltk.chat` module to import the required utilities.



```
1 import nltk
2 from nltk.chat.util import Chat, reflections
```

python

The code below shows that utility `Chat` is a class that provides logic for building the chatbot.

```
1 print(Chat)
```

python

Output:

```
1 <class 'nltk.chat.util.Chat'>
```

The other import you did above was `Reflections`, which is a dictionary that contains a set of input text and its corresponding output values. You can examine the dictionary with the code below. This is an optional dictionary and you can create your own dictionary in the same format as below.

```
1 reflections
```

python

Output:

```
1 {'i am': 'you are',
2  'i was': 'you were',
3  'i': 'you',
4  'i'm': 'you are',
5  'i'd': 'you would',
6  'i've': 'you have',
7  'i'll': 'you will',
8  'my': 'your',
9  'you are': 'I am',
```



- [Introduction](#)
- [Baseline Libraries](#)
- [Building the Chatbot](#)
- [Conclusion](#)
- [Top ^](#)



```
10     'you were': 'I was',
11     'you've': 'I have',
12     'you'll': 'I will',
13     'your': 'my',
14     'yours': 'mine',
15     'you': 'me',
16     'me': 'you'}
```

Building the Chatbot

The first step is to create rules that will be used to train the chatbot. The lines of code below create a simple set of rules. The first element of the list is the user input, whereas the second element is the response from the bot. Several such lists are created in the `set_pairs` object.

```
python
1  set_pairs = [
2      [
3          r"my name is (.*)",
4          ["Hello %1, How are you doing today
5      ],
6      [
7          r"hi|hey|hello",
8          ["Hello", "Hey there",]
9      ],
10     [
11         r"what is your name?",
12         ["You can call me a chatbot ?",]
13     ],
14     [
15         r"how are you ?",
16         ["I am fine, thank you! How can i he
17     ],
18     [
19         r"I am fine, thank you",
20         ["great to hear that, how can i help
21     ],
22     [
23         r"how can i help you?",
24         ["i am looking for online guides and
25     ],
26     [
27         r"i'm (.*) doing good",
28         ["That's great to hear", "How can i h
```

Baseline Libraries



- [Introduction](#)
- [Baseline Libraries](#)
- [Building the Chatbot](#)
- [Conclusion](#)
- [Top ^](#)



```
29         ],
30     [
31         r"i am looking for online guides and",
32         ["Pluralsight is a great option to l
33     ],
34     [
35         r"thanks for the suggestion. do they",
36         ["Yes, they have the world class bes
37     ],
38     [
39         r"(.*) thank you so much, that was h
40         ["Iam happy to help", "No problem, y
41     ],
42     [
43         r"quit",
44         ["Bye, take care. See you soon :) ", "It
45     ],
46     ]
```

After creating the pairs of rules above, we define the chatbot using the code below. The code is simple and prints a message whenever the function is invoked.

```
python
1     def chatbot():
2         print("Hi, I'm the chatbot you built"
3
4     chatbot()
```

Output:

```
1     Hi, I'm the chatbot you built
```

The next step is to instantiate the `Chat()` function containing the pairs and reflections.

```
python
1     chat = Chat(set_pairs, reflections)
2     print(chat)
```

Output:





- [Introduction](#)
- [Baseline Libraries](#)
- [Building the Chatbot](#)
- [Conclusion](#)
- [Top ^](#)



You have created a simple rule-based chatbot, and the last step is to initiate the conversation. This is done using the code below where the `converse()` function triggers the conversation.

```
1 chat.converse()  
2 if __name__ == "__main__":  
3     chatbot()
```

python

The code above will generate the following chatbox in your notebook, as shown in the image below.

Output:

```
In [3]: chatbot()  
Hi, I'm the chatbot you built  
  
In [12]: chat = Chat(pairs, reflections)  
print(chat)  
<nltk.chat.util.Chat object at 0x7f49c7689438>  
  
In [*]: chat.converse()  
if __name__ == "__main__":  
    chatbot()  
>
```

You're ready to interact with the chatbot. Start by typing a simple greeting, "hi", in the box, and you'll get the response "Hello" from the bot, as shown in the image below.

Output:



Baseline Libraries



- [Introduction](#)
- [Baseline Libraries](#)
- [Building the Chatbot](#)
- [Conclusion](#)
- [Top ^](#)



```
In [3]: chatbot()
Hi, I'm the chatbot you built

In [12]: chat = Chat(pairs, reflections)
print(chat)
<nltk.chat.util.Chat object at 0x7f49c7689438>

In [*]: chat.converse()
if __name__ == "__main__":
    chatbot()

>hi
Hello
>how are you
I am fine, thank you! How can i help you?
>
```

You can continue conversing with the chatbot and quit the conversation once you are done, as shown in the image below.

Output:

```
In [16]: chatbot()
Hi, I'm the chatbot you built

In [17]: chat = Chat(pairs, reflections)
print(chat)
<nltk.chat.util.Chat object at 0x7f49c7689208>

In [18]: chat.converse()
if __name__ == "__main__":
    chatbot()

>hi
Hey there
>how are you
I am fine, thank you! How can i help you?
>i am looking for online guides and courses to learn data science, can you suggest?
Pluralsight is a great option to learn data science. You can check their website
>thanks for the suggestion. do they have great authors and instructors?
Yes, they have the world class best authors, that is their strength;)
>thank you so much, that was helpful
Iam happy to help
>quit
It was nice talking to you. See you soon :)
Hi, I'm the chatbot you built
```

Conclusion

In this guide, you learned about creating a simple chatbot in Python. You used simple rules and the powerful `nltk` library to build the chatbot. More complex rules can be added to further strengthen the chatbot.

To learn more about text analytics and natural language processing, please refer to the following guides.



- [Introduction](#)
- [Baseline Libraries](#)
- [Building the Chatbot](#)
- [Conclusion](#)
- [Top](#) ^

1. Natural Language Processing - Machine Learning with Text Data
2. Natural Language Processing - Text Parsing
3. Natural Language Processing - Topic Identification
4. Visualizing Text Data Using a Word Cloud
5. Named Entity Recognition (NER)
6. Building Features from Text Data
7. Natural Language Processing - Extracting Sentiment from the Text Data

To learn more about data science using Python, please refer to the following guides.

1. Scikit Machine Learning
2. Linear, Lasso, and Ridge Regression with scikit-learn
3. Non-Linear Regression Trees with scikit-learn
4. Machine Learning with Neural Networks Using scikit-learn
5. Validating Machine Learning Models with scikit-learn
6. Ensemble Modeling with scikit-learn
7. Preparing Data for Modeling with scikit-learn
8. Data Science Beginners
9. Interpreting Data Using Descriptive Statistics with Python
10. Importing Data in Python
11. Finding Relationships in Data with Python