

CREATE A CHATBOT IN PYTHON

Phase-1: Document Submission

TEAM MEMBERS:

- 110521106021- Sarathy. V
- 110521106022- Sathish. S
- 11052110621- Narendhiran. C
- 110521106322- Naresh. V

OVERVIEW:

Driven by AI, automated rules, natural-language processing (NLP), and machine learning (ML), chatbots process data to deliver responses to requests of all kinds. There are two main types of chatbots. Task-oriented (declarative) chatbots are single-purpose programs that focus on performing one function. Chatbots are designed to give people an automated way to communicate with your company. They may answer basic questions, make product recommendations, and provide customer support.

DESIGN THINKING:

FUNCTIONALITY

The best chatbot has features like no code deployment, omnichannel messaging support, fallback options, sentiment analysis to add value to conversations. Integrating chatbots can help your business deliver automated smart responses and achieve marketing goals efficiently.

USER INTERFACE

A chatbot user interface (UI) is a series of graphical and language elements that allow for human-computer interaction. There are different types of user interfaces, chatbots being a natural language user interface. This means users can communicate on their terms, not the computer's.

NATURAL LANGUAGE PROCESSING (NLP)

An natural language processing chatbot is a software program that can understand and respond to human speech. Bots powered by NLP allow people to communicate with computers in a way that feels natural and human-like — mimicking person-to-person conversations.

RESPONSES

The bot displays responses in the same order you composed them. You can apply Filters to your bot responses to trigger them only when a condition is met. You can decide how fast your chatbot should respond to a user's question using the Delay feature.

INTEGRATION

Chatbot integration means you are connecting the chatbot with various platforms. It's an entire process to deploy the chatbot with various social platforms and applications.

TESTING AND IMPROVEMEN

Whenever there are updates or enhancements to the chatbot, retesting is necessary. Test the chatbot with a diverse set of user scenarios, including common queries, complex inquiries, and edge cases, to identify any vulnerabilities. When you integrate the chatbot into your software, website, or application.

Dataset Link:

<https://www.kaggle.com/datasets/grafstor/simple-dialogs-for-chatbot>

PYTHON PROGRAMMING

IN[1]:

```
import numpy as np
import string
from nltk.corpus import stopwords
import pandas as pd
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.tree import DecisionTreeClassifier
from sklearn.feature_extraction.text import TfidfTransformer, TfidfVectorizer
from sklearn.pipeline import Pipeline
```

IN[2]:

```
df = pd.read_csv('../input/simple-dialogs-for-chatbot/dialogs.txt', sep='\t')
```

IN[3]:

```
a = pd.Series(df.columns)
```

IN[4]:

```
df
```

IN[5]:

```
a = a.rename({0: df.columns[0], 1: df.columns[1]})
```

IN[6]:

```
b = {'Questions': 'Hi', 'Answers': 'hello'}
```

IN[7]:

```
c = {'Questions': 'Hello', 'Answers': 'hi'}
```

IN[8]:

```
d = {'Questions': 'how are you', 'Answers': "i'm fine. how about yourself?"}
```

IN[9]:

```
e= {'Questions':'how are you doing','Answers':"i'm fine. how about yourself?"}
```

IN[10]:

```
df = df.append(a,ignore_index=True)
```

IN[11]:

```
df.columns=['Questions','Answers']
```

IN[12]:

```
df = df.append([b,c,d,e],ignore_index=True)
```

IN[13]:

Df

OUT[13]:

	Question	Answer	
0	i'm fine. how about yourself?	i'm pretty good. thanks for asking.	
1	i'm pretty good. thanks for asking.	no problem. so how have you been?	
2	no problem. so how have you been?	i've been great. what about you?	
3	i've been great. what about you?	i've been good. i'm in school right now.	
4	i've been good. i'm in school right now.	what school do you go to?	
...

	Question	Answer	
3724	hi, how are you doing?	i'm fine. how about yourself?	
3725	Hi	hello	
3726	Hello	hi	
3727	how are you	i'm fine. how about yourself?	
3728	how are you doing	i'm fine. how about yourself?	

IN[14]:

```
df = df.append(c,ignore_index=True)
```

IN [15]:

```
df = df.append(d,ignore_index=True)
```

IN [16]:

```
df = df.append(d,ignore_index=True)
```

IN [17]:

Df

OUT[17]:

	Questions	Answers	
0	i'm fine. how about yourself?	i'm pretty good. thanks for asking.	
1	i'm pretty good. thanks for asking.	no problem. so how have you been?	

	Questions	Answers	
2	no problem. so how have you been?	i've been great. what about you?	
3	i've been great. what about you?	i've been good. i'm in school right now.	
4	i've been good. i'm in school right now.	what school do you go to?	
...	
3727	how are you	i'm fine. how about yourself?	
3728	how are you doing	i'm fine. how about yourself?	
3729	Hello	hi	
3730	how are you	i'm fine. how about yourself?	
3731	how are you	i'm fine. how about yourself?	

3732 rows x 2 columns

IN [18]:

```
def cleaner(x):
    return [a for a in (''.join([a for a in x if a not in string.punctuation])
).lower().split()]
```

IN [19]:

```
Pipe = Pipeline([
    ('bow', CountVectorizer(analyzer=cleaner)),
    ('tfidf', TfidfTransformer()),
    ('classifier', DecisionTreeClassifier())
])
```

IN [20]:

```
Pipe.fit(df['Questions'],df['Answers'])
```

OUT [20]:

```
Pipeline(steps=[('bow',  
                  CountVectorizer(analyzer=<function cleaner at 0x7f5cfaae40e0>  
)),  
                ('tfidf', TfidfTransformer()),  
                ('classifier', DecisionTreeClassifier())])
```

IN [21]:

```
Pipe.predict(['hi'])[0]
```

OUT [21]:

```
'hello'
```

IN [22]:

```
Pipe.predict(['how are you'])[0]
```

OUT [22]:

```
"i'm fine. how about yourself?"
```

IN [23]:

```
Pipe.predict(['great'])[0]
```

OUT [23]:

```
'i appreciate that.'
```

IN [24]:

```
Pipe.predict(['What are you doing'])[0]
```

OUT [24]:

```
"i'm going to change the light bulb. it burnt out."
```

CONCLUSION

A chatbot is one of the simple ways to transport data from a computer without having to think for proper keywords to look up in a search or browse several web pages to collect information; users can easily type their query in natural language and retrieve information.