

We are working on the chatbot wikipedia data

```
In [1]: #importing all the necessary libraries
import nltk
import numpy as np
import random
import string
```

```
In [2]: #opening the file from which we had to read and then making tokens
f=open('chatbot.txt','r',errors='ignore')
raw=f.read()
nltk.download('punkt')
nltk.download('wordnet')
sent_tokens=nltk.sent_tokenize(raw)
word_tokens=nltk.word_tokenize(raw)
```

```
[nltk_data] Downloading package punkt to C:\Users\Himani
[nltk_data]      Mogra\AppData\Roaming\nltk_data...
[nltk_data]      Package punkt is already up-to-date!
[nltk_data] Downloading package wordnet to C:\Users\Himani
[nltk_data]      Mogra\AppData\Roaming\nltk_data...
[nltk_data]      Package wordnet is already up-to-date!
```

```
In [3]: sent_tokens[:2]
```

```
Out[3]: ['\nChatbot\nFrom Wikipedia, the free encyclopedia\nJump to navigationJ
ump to search\nFor other uses, see Chatbot (disambiguation).',
'A virtual assistant chatbot\nELIZA conversation.jpg\nA chatbot is a s
oftware application used to conduct an on-line chat conversation via te
xt or text-to-speech, in lieu of providing direct contact with a live h
uman agent.']
```

```
In [4]: word_tokens[:2]
```

```
Out[4]: ['Chatbot', 'From']
```

```
In [5]: lemmmer=nlk.stem.WordNetLemmatizer()

def LemTokens(tokens):
    return [lemmer.lemmatize(token) for token in tokens]
remove_punc_dict=dict((ord(punc),None) for punc in string.punctuation)

def LemNormalize(text):
    return LemTokens(nltk.word_tokenize(text.lower().translate(remove_punc_dict)))
```

```
In [6]: #handling the greeting words
Greeting_input=["hello","hi","ssup","greetings","what's up","hey"]
Greeting_output=["hi","hey","nods","hey! there","hello","Hello,I'm glad,you are talking to me!"]
def greeting(sentence):
    for word in sentence.split():
        if word.lower() in Greeting_input:
            return random.choice(Greeting_output)
```

```
In [7]: from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.metrics.pairwise import cosine_similarity
```

```
In [8]: def response(user_response):
rob_response=""
sent_tokens.append(user_response)
tfidfvec=TfidfVectorizer(tokenizer=LemNormalize,stop_words="english")
tfidf=tfidfvec.fit_transform(sent_tokens)
vals=cosine_similarity(tfidf[-1],tfidf)
idx=vals.argsort()[0][-2]
flat=vals.flatten()
flat.sort()
req_tfidf=flat[-2]
if(req_tfidf==0):
    rob_response=rob_response+"I am sorry! I didn't get you"
    return rob_response
else:
```

```
rob_response=rob_response+sent_tokens[idx]
return rob_response
```

```
In [11]: flag=True
print("Chatbot : Hey There My name is TalkBot! I can help you with the
chatbots.If you want to quit to can just type Bye anytime! !")
while flag==True:
    user_response=input()
    user_response=user_response.lower()
    if user_response!="bye":
        if(user_response=="thanks" or user_response=="thank you"):
            flag=False
            print("TalkBot : You're Welcome!")
        else:
            if(greeting(user_response)!=None):
                print("TalkBot : "+greeting(user_response))
            else:
                print("TalkBot : ",end=" ")
                print(response(user_response))
                sent_tokens.remove(user_response)
    else:
        flag=False
        print("TalkBot : Thanks for talking,Bye-Bye!")
```

```
Chatbot : Hey There My name is TalkBot! I can help you with the chatbot
s.If you want to quit to can just type Bye anytime! !
```

```
hi
```

```
TalkBot : nods
```

```
hello
```

```
TalkBot : hey
```

```
ssup
```

```
TalkBot : hey
```

```
who is alan turning
```

```
TalkBot : [6]
```

```
Contents
```

```
1 Background
```

```
2      Development
3      Application
3.1    Messaging apps
3.1.1  As part of company apps and websites
3.1.2  Chatbot sequences
3.2    Company internal platforms
3.3    Customer Service
3.4    Healthcare
3.5    Politics
3.6    Toys
4      Malicious use
5      Limitations of Chatbots
6      Chatbots and jobs
7      See also
8      References
8.1    Further reading
```

Background

In 1950, Alan Turing's famous article "Computing Machinery and Intelligence" was published,[7] which proposed what is now called the Turing test as a criterion of intelligence.

thanks

TalkBot : You're Welcome!

In []: