In [1]:

```
import numpy as np
import nltk
import string
import random
```

Importing and reading the corpus

In [8]:

```
f = open('chatbot.txt','r', errors = 'ignore')
raw_doc =f.read()
raw_doc =raw_doc.lower() # convert text to lowercase
nltk.download('puntk')  #using the puntk tokenizer
nltk.download('wordnet')  #using the wordnet dictionary
sent_tokens = nltk.sent_tokenize(raw_doc)  #converts doc to list of sentences
word_tokens = nltk.word_tokenize(raw_doc)  #converts doc to list of words
```

```
[nltk_data] Error loading puntk: <urlopen error [Errno 11001]
[nltk_data]     getaddrinfo failed>
[nltk_data] Error loading wordnet: <urlopen error [Errno 11001]
[nltk_data]     getaddrinfo failed>
```

Example of sentence tokens

In [3]:

```
sent_tokens[:2]
```

Out[3]:

```
['data science is an interdisciplinary field that uses scientific methods, processes, algorithms and systems to extract knowledge and insights from noisy, structured and unstructur
ed data,[1][2] and apply knowledge from data across a broad range of application domains.',
 'data science is related to data mining, machine learning and big data.']
```

Example of word tokens

In [4]:

```
word_tokens[:2]
```

Out[4]:

```
['data', 'science']
```

Text preprocessing

In [5]:

```
lemmer = nltk.stem.WordNetLemmatizer()

# wordnet is A  semantically-oriented dictionary of english included in NLTK.

def LemTokens(tokens):
    return [lemmer.lemmatize(token) for token in tokens]
remove_punct_dict = dict((ord(punct), None) for punct in string.punctuation)
def LemNormalize(text):
    return LemTokens(nltk.word_tokenize(text.lower().translate(remove_punct_dict)))
```

defining the greeting function

In [6]:

```
GREET_INPUTS =("hello","hi","greeting","sup","what's up","hey",)
GREET_RESPONSES = ("hi","hey","hi there","hello","I am glad! You are talking to me.")
def greet(sentences):

    for word in sentences.split():
        if word.lower()in  GREET_INPUTS:
            return random.choice(GREET_RESPONSES)
```

Response generation

In [7]:

```
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.metrics.pairwise import cosine_similarity

def response (user_response):
    robo1_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):
        robo1_response = robo1_response+"I am sorry! I don't understand you."
        return robo1_response
    else:
        robo1_response = robo1_response+sent_tokens[idx]
        return robo1_response
```

defining conversation start/end protocals

In [9]:

```
flag=True
print("BOT: My name is stark. Let's have a conversation! Also, if you want to exit any time, just type Bye.")
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("BOT: you are welcome.")
        else:
            if(greet(user_response)):
                print("BOT: "+greet(user_response))
            else:
                sent_tokens.append(user_response)
                word_tokens=word_tokens+nltk.word_tokenize(user_response )
                final_words=list(set(user_response))
                print("BOT: ",end="")
                print(response(user_response))
                sent_tokens.remove(user_response)
    else:
        flag=False
        print("BOT: Goodbye! Take care <3")
```

```
BOT: My name is stark. Let's have a conversation! Also, if you want to exit any time, just type Bye.
hi
BOT: I am glad! You are talking to me.
how are you
BOT:
C:\Users\ckhan\anaconda3\lib\site-packages\sklearn\feature_extraction\text.py:388: UserWarning: Your stop_words may be inconsistent with your preprocessing. Tokenizing the stop wor
ds generated tokens ['ha', 'le', 'u', 'wa'] not in stop_words.
  warnings.warn('Your stop_words may be inconsistent with '
I am sorry! I don't understand you.
foundation
BOT: [7]


contents
1       foundations
1.1     relationship to statistics
2       etymology
2.1     early usage
2.2     modern usage
3       see also
4       references
foundations
data science is an interdisciplinary field focused on extracting knowledge from typically large data sets and applying the knowledge and insights from that data to solve problems i
n a wide range of application domains.
early usage
BOT: etymology
early usage
in 1962, john tukey described a field he called "data analysis", which resembles modern data science.
references
BOT: [31]

see also
international journal of population data science
references
 dhar, v. (2013).
bye
BOT: Goodbye! Take care <3
```

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]: