Aim:
Implement Text Similarity Recognizer for chosen text document. Build a chatbot for Agriculture Domain.

Theory :
A chatbot is a software program for simulating intelligent conversations with human using rules or artificial intelligence. Users interact with the chatbot via conversational interface through written or spoken text. Chatbots can live in messaging platforms like Slack, Facebook Messenger and Telegram and serve many purposes – ordering products, knowing about weather and managing your finance among other things.

Limitations With A Chatbot

With increasing advancements, there also comes a point where it becomes fairly difficult to work with the chatbots. Following are a few limitations we face with the chatbots.

- **Domain Knowledge** – Since true artificial intelligence is still out of reach, it becomes difficult for any chatbot to completely fathom the conversational boundaries when it comes to conversing with a human.
- **Personality** – Not being able to respond correctly and fairly poor comprehension skills has been more than frequent errors of any chatbot, adding a personality to a chatbot is still a benchmark that seems far far away.

We can define the chatbots into two categories, following are the two categories of chatbots:

1. **Rule-Based Approach** – In this approach, a bot is trained according to rules. Based on this a bot can answer simple queries but sometimes fails to answer complex queries.
2. **Self-Learning Approach –** These bots follow the machine learning approach which is rather more efficient and is further divided into two more categories.
    - **Retrieval-Based Models** – In this approach, the bot retrieves the best response from a list of responses according to the user input.
    - **Generative Models** – These models often come up with answers than searching from a set of answers which makes them intelligent bots as well.

Building a chat bot requires it to have a corpus of text to serve as his knowledge and a mechanism to be able to connect the questions asked with the knowledge. One of the most easiest way to do so is by using Co-sine similarity.
Cosine Similarity :
Cosine similarity is a measure of similarity between two non-zero vectors of an inner product space that measures the cosine of the angle between them. The cosine of 0° is 1, and it is less than 1 for any angle in the interval (0, π] radians. It is thus a judgment of orientation and not magnitude: two vectors with the same orientation have a cosine similarity of 1, two vectors oriented at 90° relative to each other have a similarity of 0, and two vectors diametrically opposed have a similarity of -1, independent of their

magnitude. The cosine similarity is particularly used in positive space, where the outcome is neatly bounded . The name derives from the term "direction cosine": in this case, unit vectors are maximally "similar" if they're parallel and maximally "dissimilar" if they're orthogonal (perpendicular). This is analogous to the cosine, which is unity (maximum value) when the segments subtend a zero angle and zero (uncorrelated) when the segments are perpendicular.

$$\text{similarity} = \cos(\theta) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\|\|\mathbf{B}\|} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2}\sqrt{\sum_{i=1}^{n} B_i^2}}$$

To calculate the cosine similarity between what is asked to the chat bot and the corpus yext which is the knowledge , text is converted into tokens using TF-IDF Vectorizer .Tf–idf or TFIDF, short for term frequency–inverse document frequency, is a numerical statistic that is intended to reflect how important a word is to a document in a collection or corpus.It is often used as a weighting factor in searches of information retrieval, text mining, and user modeling. The tf–idf value increases proportionally to the number of times a word appears in the document and is offset by the number of documents in the corpus that contain the word, which helps to adjust for the fact that some words appear more frequently in general.

Conclusion :

Thus , we were able to create a chatbot for agriculture related questions using Cosine similarity .

Code :

```
import spacy
import numpy as np
import random
import string
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.metrics.pairwise import x
import nltk
from spacy import displacy
from collections import Counter
import en_core_web_sm
nlp = en_core_web_sm.load()
from google.colab import files
uploaded = files.upload()
f=open('agri.txt','r',errors = 'ignore')
raw=f.read()
raw=raw.lower()# converts to lowercase
nltk.download('punkt') # first-time use only
nltk.download('wordnet') # first-time use only
sent_tokens = nltk.sent_tokenize(raw)# converts to list of sentences
word_tokens = nltk.word_tokenize(raw)# converts to list of words


lemmer = nltk.stem.WordNetLemmatizer()
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)))
GREETING_INPUTS = ("hello", "hi", "greetings", "sup", "what's up","hey","yo")
GREETING_RESPONSES = ["hi", "hey", "*nods*", "hi there", "hello", "I am glad! You are talking to
me"]
def greeting(sentence):
  for word in sentence.split():
    if word.lower() in GREETING_INPUTS:
      return random.choice(GREETING_RESPONSES)

    def response(user_response):
      robo_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):
        robo_response=robo_response+"I am sorry! I don't understand you"
        return robo_response
```

```
  else:
    robo_response = robo_response+sent_tokens[idx]
    return robo_response
flag=True
print("ROBO: My name is Robo. I will answer your queries about Chatbots. If you want to exit,
        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("ROBO: You are welcome..")
    else:
      if(greeting(user_response)!=None):
        print("ROBO: "+greeting(user_response))
      else:
        print("ROBO: ",end="")
        print(response(user_response))
        sent_tokens.remove(user_response)
  else:
    flag=False
    print("ROBO: Bye! take care..")
```

Output :

```
ROBO: My name is Robo. I will answer your queries about Chatbots. If you want to exit, type Bye!
hey
ROBO: I am glad! You are talking to me
what is agriculture
ROBO: industrial agriculture based on large-scale monoculture in the twentieth century came to
        dominate agricultural output, though about 2 billion people still depended on subsistence
        agriculture into the twenty-first.
what is land
ROBO: estimates of the amount of land transformed by humans vary from 39 to 50%.land
        degradation, the long-term decline in ecosystem function and productivity, is estimated to be
        occurring on 24% of land worldwide, with cropland overrepresented.the un-fao report cites
        land management as the driving factor behind degradation and reports that 1.5 billion people
        rely upon the degrading land.
bye
ROBO: Bye! take care..
```