

CHATBOT

Team Name: **Noob Entity**

Team Number: **06**

M Jaipal (9919004177)

S Likhil Srinivas (9919004269)

H Nikhil Reddy (9919004106)

S V Nagendra (9919004270)

ABSTRACT:

A chatbot is software that simulates human-like conversations with users via text messages on chat. Its key task is to help users by providing answers to their questions. It can simulate a conversation (or a chat) with a user in natural language through messaging applications, websites, mobile apps or through the telephone. A chatbot is often described as one of the most advanced and promising expressions of interaction between humans and machines. However, from a technological point of view, a chatbot only represents the natural evolution of a Question Answering system leveraging Natural Language Processing (NLP). Formulating responses to questions in natural language is one of the most typical Examples of Natural Language Processing applied in various enterprises' end-use applications.

Education chatbot simply works for students and facilitators ease, helps to increase students' engagement towards an institute or their studies, and provides instant support as a facilitator. If we request the certain keywords or definitions from the Input file our chat bot searches the document and give the accurate meaning or definition to the given keyword. If the word is not present in the input file it simply says I don't understand you. The use of chatbots in educational mobile apps help students in getting instant replies and help for their queries. Chatbots help in automating trivial tasks for students such as submitting assignments, sending replies to emails, sending instant messages, and feedback. ... Students can get instant answers to their queries.

INTRODUCTION:

Chatbot applications streamline interactions between people and services, enhancing customer experience. At the same time, they offer companies new opportunities to improve the customers engagement process and operational efficiency by reducing the typical cost of customer service.

To be successful, a chatbot solution should be able to effectively perform both of these tasks. Human support plays a key role here: Regardless of the kind of approach and the platform, human intervention is crucial in configuring, training and optimizing the chatbot system.

There are two different tasks at the core of a chatbot:

1. User request analysis
2. Returning the response

User Request Analysis:

This is the first task that a chatbot performs. It analyses the user's request to identify the user intent and to extract relevant entities.

How a Chatbot Works: Example of user request analysis.

The ability to identify the user's intent and extract data and relevant entities contained in the user's request is the first condition and the most relevant step at the core of a chatbot: If you are not able to correctly understand the user's request, you won't be able to provide the correct answer.

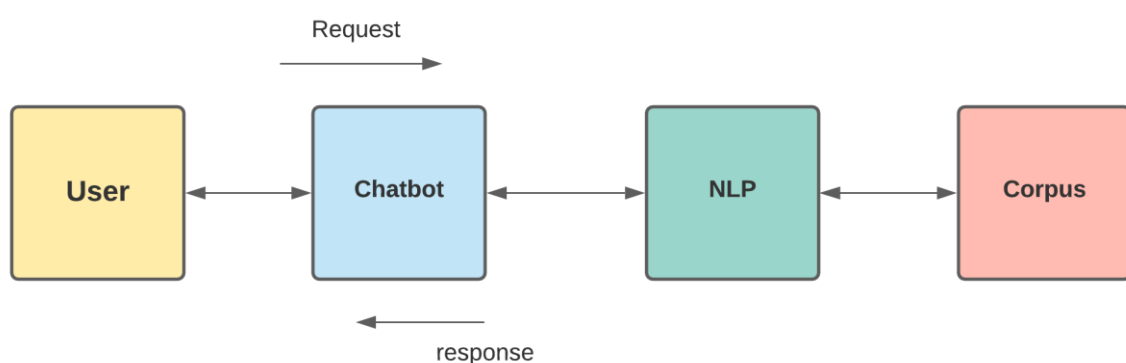
Returning the response:

Once the user's intent has been identified, the chatbot must provide the most appropriate response for the user's request. The answer may be:

- a generic and predefined text;
- a text retrieved from a knowledge base that contains different answers;
- a contextualized piece of information based on data the user has provided;
- data stored in enterprise systems;
- the result of an action that the chatbot performed by interacting with one or more backend application.
- a disambiguating question that helps the chatbot to correctly understand the user's request.

Chatbot applications streamline interactions between people and services, enhancing customer experience. At the same time, they offer companies new opportunities to improve the customers engagement process and operational efficiency by reducing the typical cost of customer service. Chatbots based on Artificial Intelligence (AI) and Natural Language Processing (NLP). These bots are intelligent and can “learn” over time to respond like humans would to free-form user inputs.

ARCHITECTURE:



METHODOLOGY:

I. Importing libraries

numpy – python library used for working with arrays.

nltk – nlp library used for building programs that works with human language data.

string – helps to process and handle strings

random – helps to choose random response

II. Reading the corpus

We have to bring and read the corpus/input file. convert corpus text into lowercase and download punkt tokenizer - which is a pretrained tokenizer has a complete capability to build model around what's required and gives an answer .

Wordnet – database for English language helps to find meanings of words, synonyms, antonyms and more.

Split the document into list of sentences and words.

```
[nltk_data] Downloading package punkt to /root/nltk_data...  
[nltk_data] Unzipping tokenizers/punkt.zip.  
[nltk_data] Downloading package wordnet to /root/nltk_data...  
[nltk_data] Unzipping corpora/wordnet.zip.
```

III. Text Processing

Process the text by removing punctuation marks from list of sentences and words.

IV. Defining greeting functions

Define list of greet_inputs and greet_responses . So when user gives any one of the input from greet_input, return a random choice from greet_responses.

V. Response generation

Import tfidfVectorizer – basically stands for term frequency(maps how many times a individual words repeated in our corpus) and Inverse document frequency(mapping how rare a particular word is present in our corpus).

Import cosine_similarity – helps to get normalize output so machine can understand what we are doing.

Define a situation where let user know when chatbot doesn't understood what user is requesting.

VI. Defining conversation start/end protocols

Loop the request and reponse process until user says bye.

If response is found return response else return chatbot doesn't understood.

```
Bot: My name is Stark. Let's have a conversation! Also, if you want to exit any time, just type Bye!
hey
BOT:I am glad! You are talking to me
foundations
BOT: /usr/local/lib/python3.7/dist-packages/sklearn/feature_extraction/text.py:385: UserWarning: Your
'stop_words.' % sorted(inconsistent))
[6]

contents
1      foundations
1.1    relationship to statistics
2      etymology
2.1    early usage
2.2    modern usage
3      market
4      technologies and techniques
4.1    techniques
5      see also
6      references
foundations
data science is an interdisciplinary field focused on extracting knowledge from data sets, which are
kalasalingam
/usr/local/lib/python3.7/dist-packages/sklearn/feature_extraction/text.py:385: UserWarning: Your stop
'stop_words.' % sorted(inconsistent))
BOT: I am sorry! I don't understand you
bye
BOT: Goodbye! Take care 🍷
```

SOURCE CODE:

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

```
f=open('chatbot2.txt','r',errors = 'ignore')
raw_doc=f.read()
raw_doc=raw_doc.lower()
nltk.download('punkt')
nltk.download('wordnet')
sent_tokens = nltk.sent_tokenize(raw_doc)
word_tokens = nltk.word_tokenize(raw_doc)
```

```
sent_tokens[:2]
```

```
word_tokens[:2]
```

```
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)))
```

```
GREET_INPUTS=["hello", "hi", "greetings", "sup", "what's up","hey"]
GREET_RESPONSES = ["hi", "hey", "hi there", "hello", "I am glad! You are talking t
o me"]
```

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

```
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

```

```

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)!=None):
                print("BOT:"+greet(user_response))
            else:
                sent_tokens.append(user_response)
                word_tokens=word_tokens+nlTK.word_tokenize(user_response)
                final_words=list(set(word_tokens))
                print("BOT: ",end="")
                print(response(user_response))
                sent_tokens.remove(user_response)
            else:
                flag=False
                print("BOT: Goodbye! Take care 🍷")

```

CONCLUSION:

From my perspective, chatbots or smart assistants with artificial intelligence are dramatically changing businesses. Chatbots can reach out to a large audience on messaging apps and be more effective than humans. They may develop into a capable information-gathering tool in the near future. There is a wide range of chatbot building platforms that are available for various enterprises, such as e-commerce, retail, banking, leisure, travel, healthcare, and so on. This chatbot can answer queries in the textual user input.

At last, we can conclude that our chatbot helps students to get deeper knowledge on any particular topic.