



VIT[®]
Vellore Institute of Technology
(Deemed to be University under section 3 of UGC Act, 1956)

CHATBOT

Submitted in partial fulfillment of the requirements for the degree of

Bachelor of Technology
in
COMPUTER SCIENCE
By

Yashas M

(20BCE2607)

Sujit S Bagdure

(20BDS0098)

Kate Harshal Sanjay

(20BDS0096)

Under the guidance of Gunavathi C

SCOPE

VIT, Vellore.

DECEMBER, 2021

C1+TC1

VL2021220104514

CERTIFICATE

This is to certify that the thesis entitled “CHATBOT” submitted by

Yashas M 20BCE2607

Sujit S Bagdure 20BDS0098

Kate Harshal Sanjay 20BDS0096

, VIT, for the award of the degree of **Bachelor of Technology in Computer Science**

Engineering, is a record of bonafide work carried out by them under my supervision during the period, 01. 8. 2021 to 10.12.2021 as per the VIT code of academic and research ethics.

The contents of this report have not been submitted and will not be submitted either in part or in full, for the award of any other degree or diploma in this institute or any other institute or university. The thesis fulfills the requirements and regulations of the University and in my opinion meets the necessary standards for submission.

PLACE: VELLORE

DATE: 01-DEC-2021

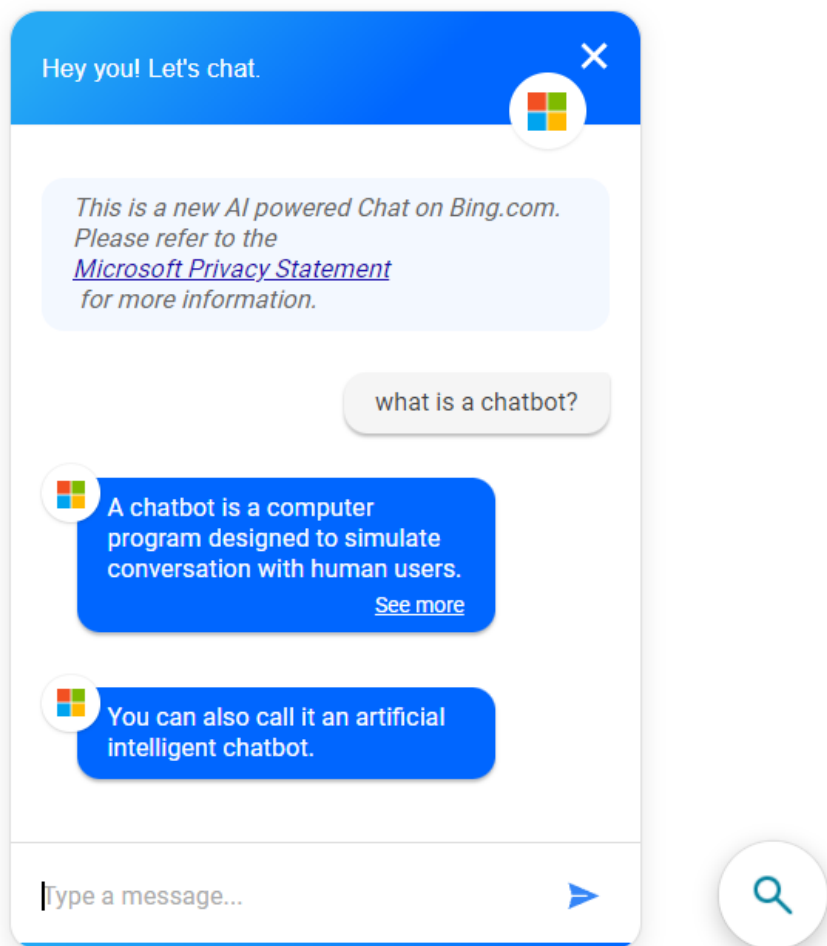
Signature of the candidate

Contents:

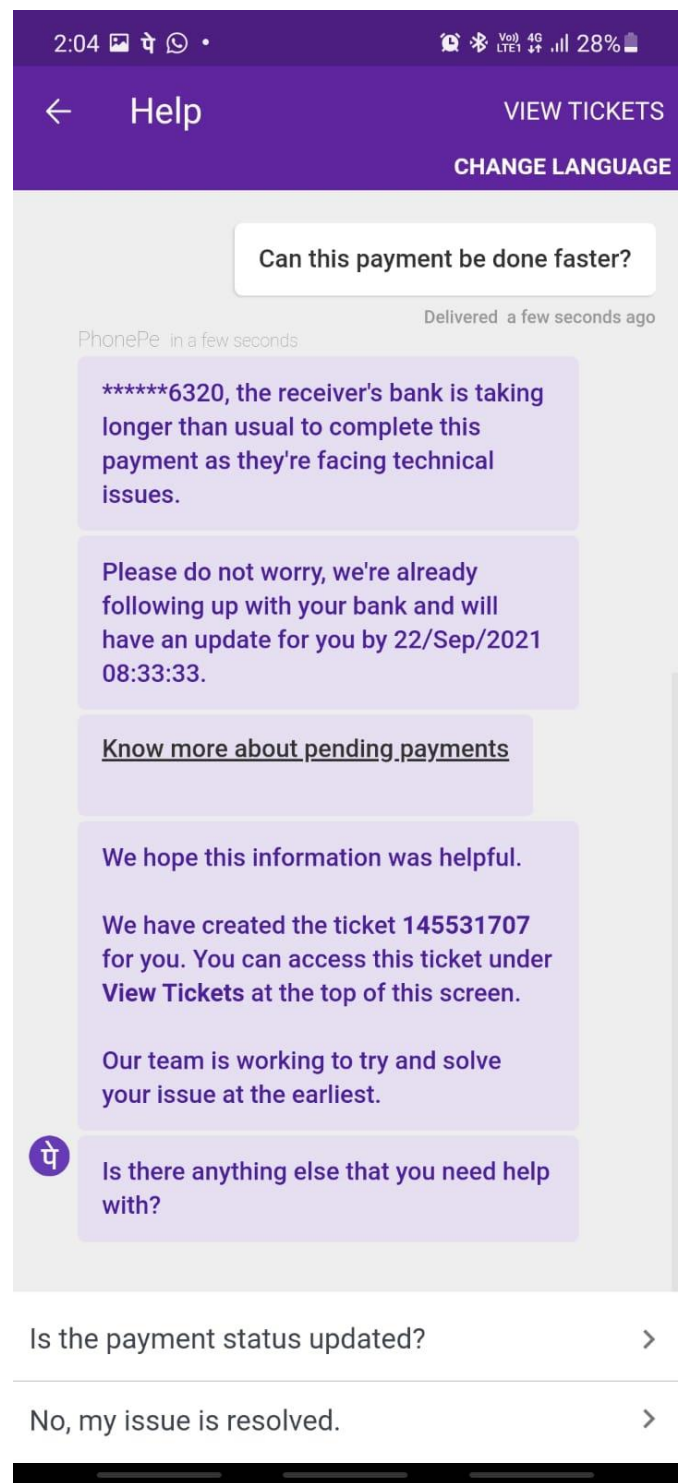
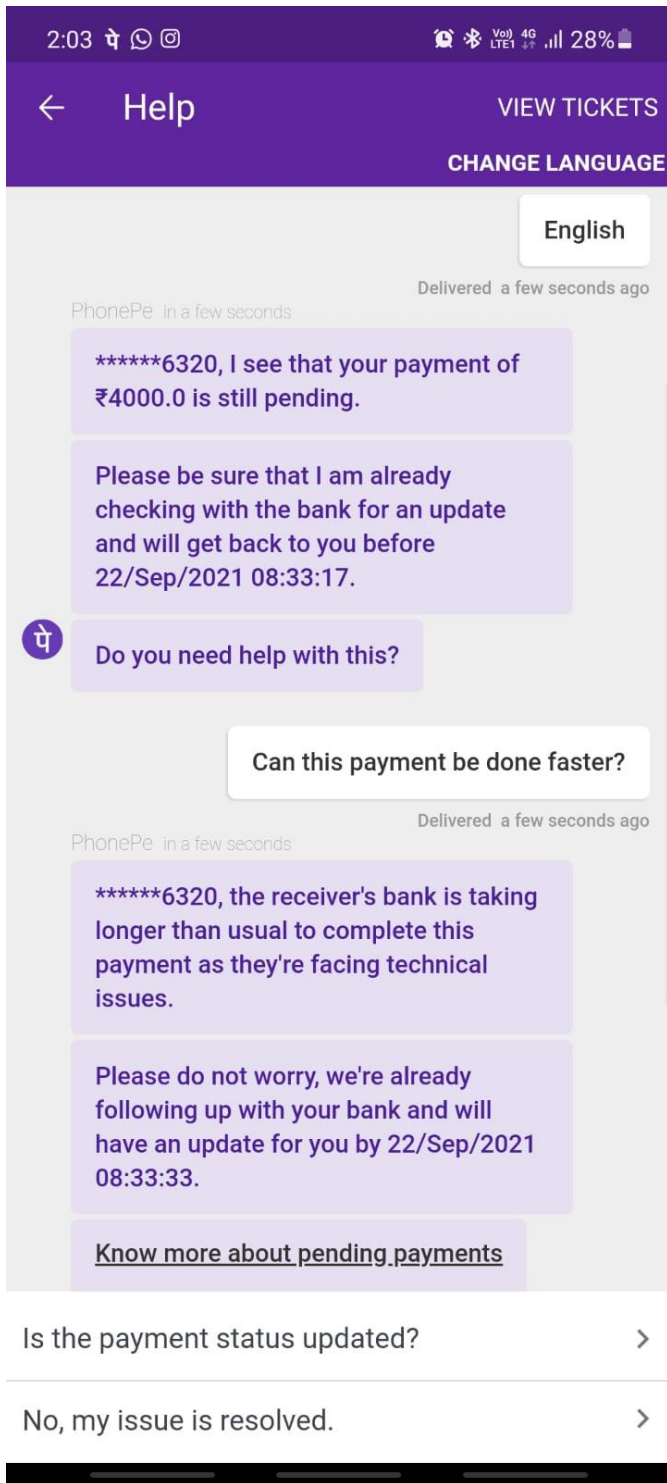
1. What is a Chatbot?
2. How do Chatbots Work?
 - a. AI-based chatbots
 - b. Rule-based chatbots
3. What is Chatbot Architecture?
 - a. Question and Answer System
 - b. Environment
 - c. Front-End Systems
 - d. Node Server / Traffic Server
 - e. Custom Integrations
4. How do Chatbots Work?
 - a. Pattern Matchers
 - b. Algorithms
 - c. Artificial Neural Networks
5. Algorithms used
 - a. NLU (NATURAL LANGUAGE UNDERSTANDING)
 - b. NLP (NATURAL LANGUAGE PROCESSING)
- 6. Project demonstration and Source Code**
 - a. AI-based chatbots
 - b. Rule-based chatbots
7. Datasets used
8. References

What is a Chatbot?

- A chatbot can be defined as a developed program capable of having a discussion/conversation with a human. Any user might, for example, ask the bot a question or make a statement, and the bot would answer or perform an action as necessary. A chatbot communicates similarly to instant messaging.
- Chatbots are widely in use now-a-days for various business or personal purpose. They brought a new way for businesses to communicate with their customers with the help of emerging technologies like Artificial Intelligence. Not only chatbots can be used for customer support but also it can help users to act as their companion.



REAL-LIFE EXAMPLE OF CHATBOT (CUSTOMER SERVICE)

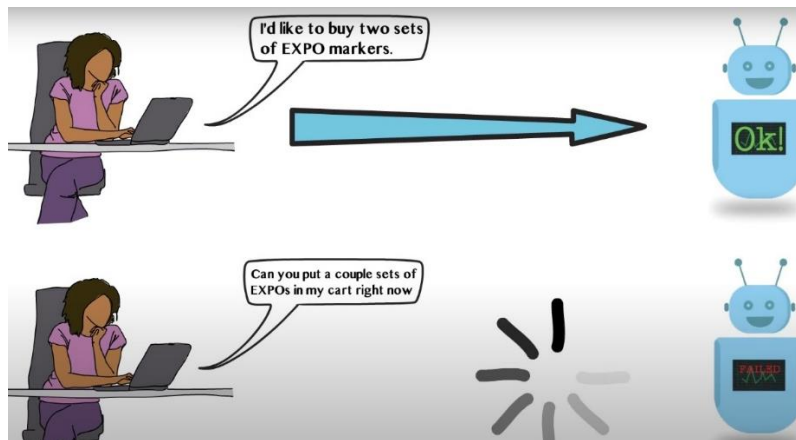


TYPES OF CHATBOTS

There are 2 categories of chatbots:

1. Rule-based chatbots:

A rule-based bot can only comprehend a limited range of choices that it has been programmed with. Predefined rules define the course of the bot's conversation. Rule-based chatbots are easier to build as they use a simple true-false algorithm to understand user queries and provide relevant answers.



2. AI-based chatbots:

This bot is equipped with an artificial brain, also known as artificial intelligence. It is trained using machine-learning algorithms and can understand open-ended queries. Not only does it comprehend orders, but it also understands the language. As the bot learns from the interactions it has with users, it continues to improve. The AI chatbot identifies the language, context, and intent, which then reacts accordingly.



What is Chatbot Architecture?

Chatbot architecture is the spine of the chatbot. The type of architecture for your chatbot depends on various factors like use-case, domain, chatbot type, etc. However, the basic conversation flow remains the same.

1. Question and Answer System:

The Q&A system is responsible for answering customers' frequently asked questions.

2. Environment:

The environment is mainly responsible for contextualizing users' messages using natural language processing (NLP). The NLP Engine is the central component of the chatbot architecture. It interprets what users are saying at any given time and turns it into organized inputs that the system can process.

3. Front-End Systems:

Front-end systems are the ones where users interact with the chatbot. These are client-facing systems such as – Facebook Messenger, WhatsApp business, Slack, Google Hangouts, your website or mobile app, etc.

4. Node Server / Traffic Server:

It is the server that deals with user traffic requests and routes them to the proper components. The response from internal components is often routed via the traffic server to the front-end systems.

5. Custom Integrations

With custom integrations, your chatbot can be integrated with your existing backend systems like CRM, database, payment apps, calendar, and many such tools, to enhance the capabilities of your chatbot.

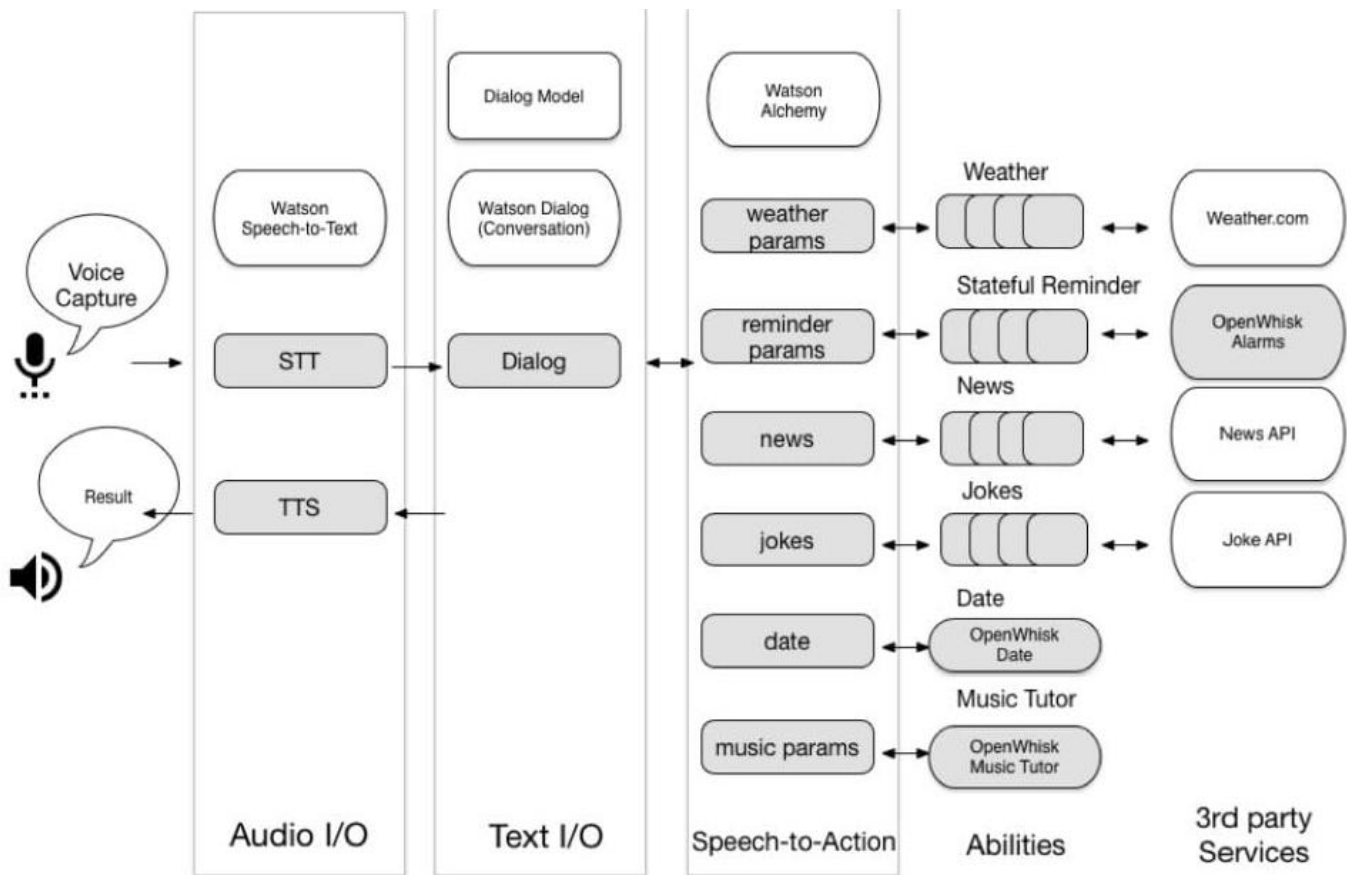
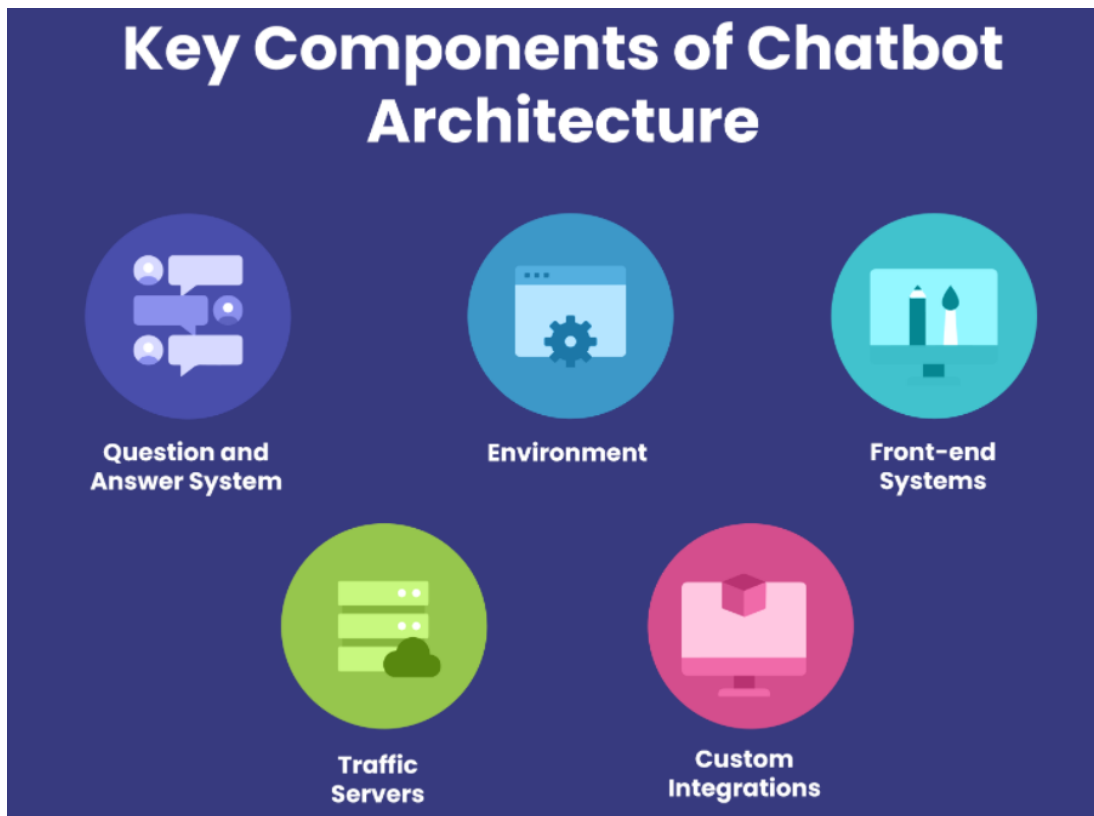


Figure 2. Chatbot Architecture

WORKING OF CHATBOTS

1. Pattern Matchers

Bots use pattern matching to classify the text and produce a suitable response for the customers. A standard structure of these patterns is “Artificial Intelligence Markup Language” (AIML).

Example:

```
<aiml version = "1.0.1" encoding = "UTF-8"?>
  <category>
    <pattern> WHO IS ABRAHAM LINCOLN </pattern>
    <template>Abraham Lincoln was the US President during American civil war.</template>
  </category>

  <category>
    <pattern>DO YOU KNOW WHO * IS</pattern>
    <template>
      <srai>WHO IS <star/></srai>
    </template>
  </category>
</aiml>
```

The machine then gives and output:

Human: Do you know who Abraham Lincoln is?

Robot: Abraham Lincoln was the US President during the American civil war.

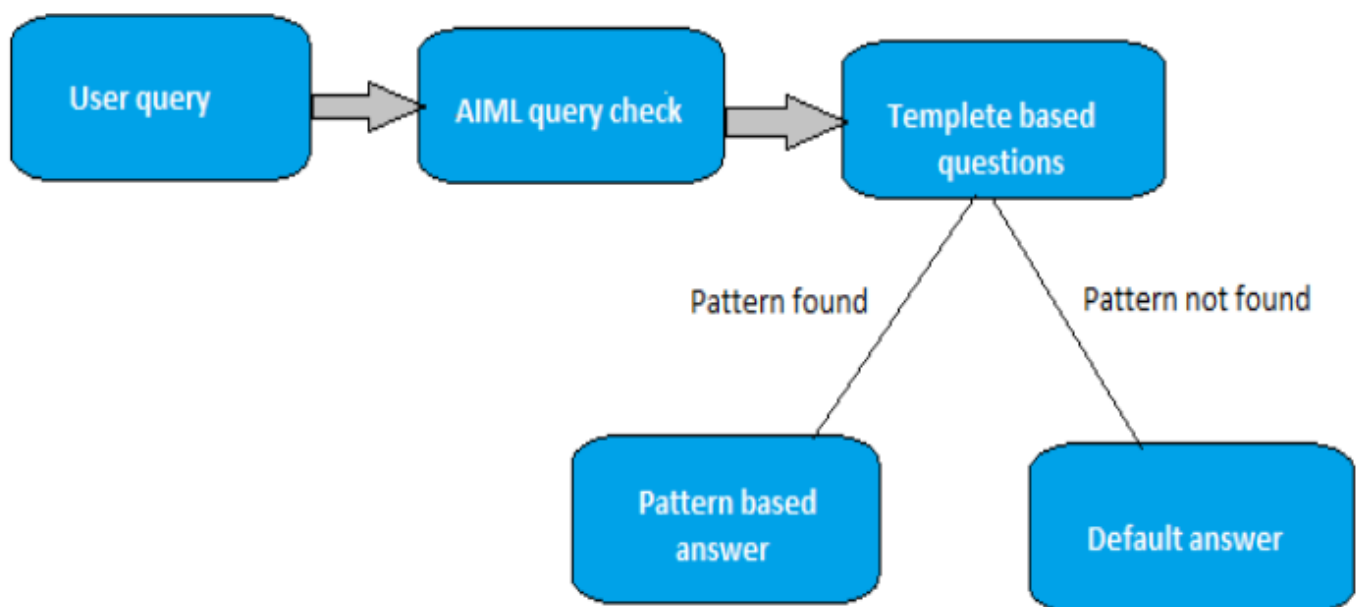
Chatbot knows the answer only because his or her name is in the associated pattern. Similarly, chatbots respond to anything relating it to the associated patterns. But it cannot go beyond the related pattern. Algorithms can help for an advanced level of working.

2. Algorithms

A unique pattern must be available in the database to provide a suitable response for each kind of question. A hierarchy is created with lots of combinations of patterns. Algorithms are used to reduce the number of classifiers and create a more manageable structure.

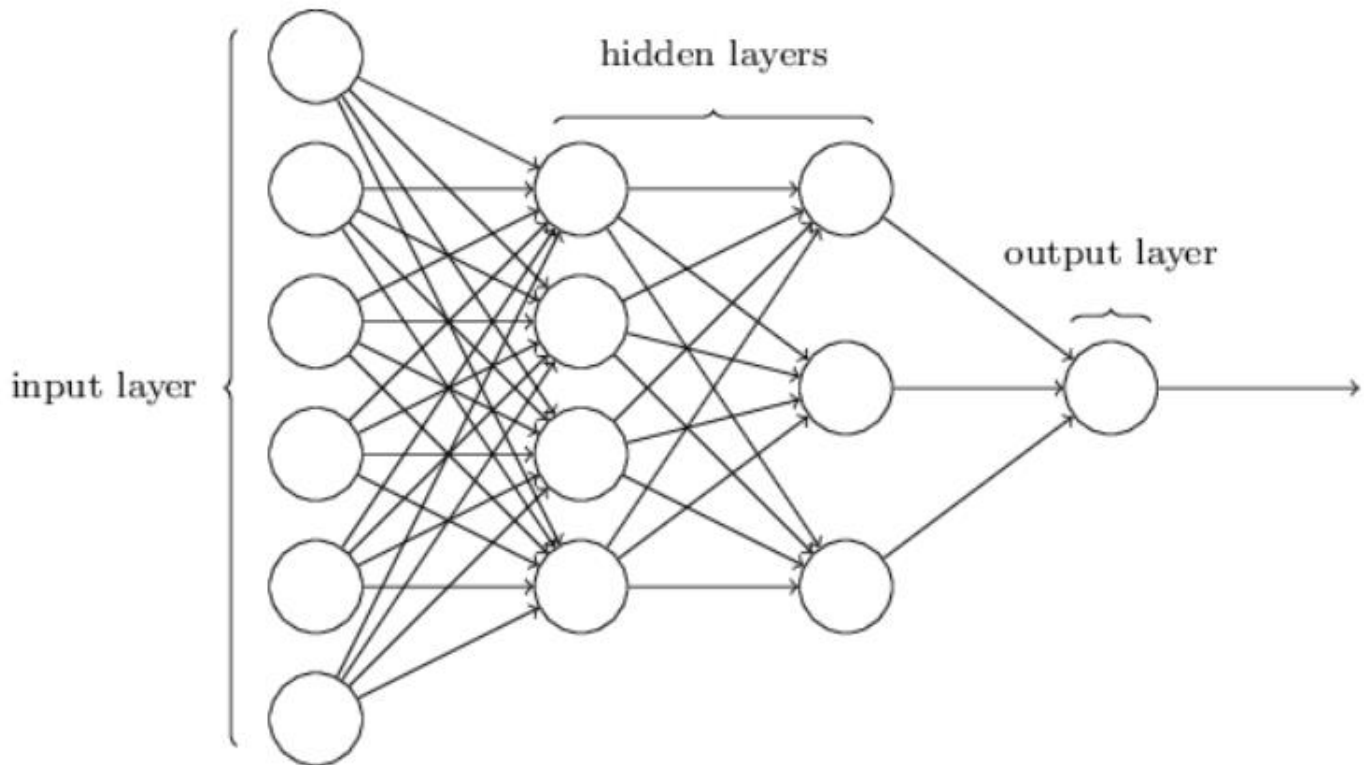
Computer scientists call it a “Reductionist” approach- to give a simplified solution; it reduces the problem.

Multinational Naive Bayes is the best example of the algorithm for NLP and text classification. For instance, let’s look at the set of sentences that belong to a particular class. With new input sentences, each word is counted for its occurrence and is accounted for its commonality. Then, each class is assigned a score. The highest scored class is the most likely to be associated with the input sentence.



3. Artificial Neural Networks

Neural Networks are a way of calculating the output from the input using weighted connections, which are computed from repeated iterations while training the data. Each step through the training data amends the weights resulting in the output with accuracy.



The trained data of a neural network is a comparable algorithm with more and less code. When there is a comparably small sample, where the training sentences have 200 different words and 20 classes, that would be a matrix of 200×20 . But this matrix size increases by n times more gradually and can cause a massive number of errors. In this kind of scenario, processing speed should be considerably high.

There are multiple variations in neural networks, algorithms as well as patterns matching code. Complexity may also increase in some of the variations. But the fundamental remains the same, and the critical work is that of classification.

NLU (NATURAL LANGUAGE UNDERSTANDING)

NLU helps the chatbot understand the query by breaking it down.

- **Entities:** An entity represents keywords from the user's query picked up by the chatbot to understand what the user wants.
- **Intents:** It helps identify the action the chatbot needs to perform on the user's input.
- **Context:** With context, you can easily relate intents without any need to know what was the previous question.



ARTIFICIAL INTELLIGENCE MARKUP LANGUAGE

It has class of data object called an AIML object that describes the behavior of computer programs. It consists of units or tag called topics and categories. In AIML, categories are basic unit of knowledge. Each category consists of pattern which contains input and template which contain answer of chatbot.

There are three types of AIML classes:

Atomic category:

```
< category >  
< pattern >How are you< /pattern >  
< template >I am fine!< /template >  
< /category >
```

Default category:

```
< category >  
< pattern >Who is * < /pattern >  
< template > He is my brother < /template >  
< /category >
```

Recursive category:

```
< category >  
< pattern > Can you tell who the * is < /pattern >  
< template > He is my brother  
< srai > Who is * < /srai >  
< /template >  
< /category >
```

NLP (NATURAL LANGUAGE PROCESSING)

Natural Language Processing (NLP) chatbot takes some steps to convert the customer's text or speech into structured data to select the related answer.

- **Sentiment Analysis:** With this, the algorithm tries to interpret the sentiment of the user's query by reading into the entities, themes, and topics.
- **Tokenization:** The NLP divides a string of words into pieces or tokens. These tokens are linguistically symbolic or are differently helpful for the application.
- **Named Entity Recognition:** The chatbot program model looks for categories of words, like the name of the product, the user's name or address, whichever data is required.
- **Normalization:** The chatbot program model processes the text to find common spelling mistakes or typographical errors in the user's intent. It gives a more human-like effect of the chatbot to the users.
- **Dependency Parsing:** The chatbot looks for the objects and subjects- verbs, nouns and common phrases in the user's text to find dependent and related terms that users might be trying to convey.

Datasets for the AI based chatbot

```
chatbot - Notepad
File Edit Format View Help
Data science

Data science is an interdisciplinary field that uses scientific methods, processes, algorithms and systems to extract knowledge and insights from noisy, structured and unstructured data or data sets.

Data science is a "concept to unify statistics, data analysis, informatics, and their related methods" in order to "understand and analyze actual phenomena" in the context of the real world.

A data scientist is someone who creates programming code, and combines it with statistical knowledge to create insights from data.[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 typically large (see big data), and applying the knowledge and methods of the three emerging foundational professional communities.[12]

Relationship to statistics
Many statisticians, including Nate Silver, have argued that data science is not a new field, but rather another name for statistics.[13] Others argue that data science is a new field.

Etymology
Early usage
In 1962, John Tukey described a field he called "data analysis", which resembles modern data science.[17] In 1985, in a lecture given to the Chinese Academy of Sciences, he described "data science" as a field that combines statistics, computer science, and operations research.

The term "data science" has been traced back to 1974, when Peter Naur proposed it as an alternative name for computer science.[21] In 1996, the International Federation of Data Science (IFDS) was founded.

During the 1990s, popular terms for the process of finding patterns in datasets (which were increasingly large) included "knowledge discovery" and "data mining".

Ln 1, Col 1 100% Windows (CRLF) UTF-8
```

```
chatbot - Notepad
File Edit Format View Help
Many statisticians, including Nate Silver, have argued that data science is not a new field, but rather another name for statistics.[13] Others argue that data science is a new field.

Etymology
Early usage
In 1962, John Tukey described a field he called "data analysis", which resembles modern data science.[17] In 1985, in a lecture given to the Chinese Academy of Sciences, he described "data science" as a field that combines statistics, computer science, and operations research.

The term "data science" has been traced back to 1974, when Peter Naur proposed it as an alternative name for computer science.[21] In 1996, the International Federation of Data Science (IFDS) was founded.

During the 1990s, popular terms for the process of finding patterns in datasets (which were increasingly large) included "knowledge discovery" and "data mining".

Modern usage
The modern conception of data science as an independent discipline is sometimes attributed to William S. Cleveland.[24] In a 2001 paper, he advocated an expansion of the traditional statistical approach to include computer science and operations research.

The professional title of "data scientist" has been attributed to DJ Patil and Jeff Hammerbacher in 2008.[26] Though it was used by the National Science Board in 1987, it was not widely used until the late 2000s.

There is still no consensus on the definition of data science and it is considered by some to be a buzzword.[28]

Market
Big data is becoming a tool for businesses and companies of all sizes.[29] The availability and interpretation of big data has altered the business models of many industries.

Technologies and techniques
There is a variety of different technologies and techniques that are used for data science which depend on the application. More recently, full-featured, end-to-end data science platforms have emerged.

Techniques
Further information: Statistics § Methods
Linear regression
Logistic regression
Decision trees are used as prediction models for classification and data fitting. The decision tree structure can be used to generate rules able to classify data.
Support-vector machine (SVM)
Cluster analysis is a technique used to group data together.
Dimensionality reduction is used to reduce the complexity of data computation so that it can be performed more quickly.
Machine learning is a technique used to perform tasks by inferring patterns from data.
Naive Bayes classifiers are used to classify by applying the Bayes' theorem. They are mainly used in datasets with large amounts of data, and can aptly generate predictions.

Ln 1, Col 1 100% Windows (CRLF) UTF-8
```

AI – BASED CHATBOT

CODE:

Untitled0.ipynb

File Edit View Insert Runtime Tools Help

+ Code + Text

```
✓ [1] import numpy as np
1s    import nltk
      import string
      import random
```

importing and reading the corpus

```
✓ [3] f=open('chatbot.txt','r',errors='ignore')
2s    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)
```

```
[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.
```

example of sentence tokens

```
[ ] sent_tokens[:2]
```

```
['data science\n\ndata science is an interdisciplinary field that uses scientific methods, pro
'data science is related to data mining, machine learning and big data.']
```

example of word tokens

```
[ ] word_tokens[:2]
```

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

text preprocessing

```
[ ] lemmer=nlk.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)))
```

defining the greeting function

```
▶ GREET_INPUTS = ("hello", "hi", "sup", "greetings", "what's up", "hey")
GREET_RESPONSES = ["hi", "hey", "*nods*", "hi there", "i'm glad you are talking to me"]
def greet(sentence):
    for word in sentence.split():
        if word.lower() in GREET_INPUTS:
            return random.choice(GREET_RESPONSES)
```

response generation

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

[ ] def response(robol_response):
    robol_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):
        robol_response=robol_response+"i'm sorry! i don't understand you"
        return robol_response
    else:
        robol_response=robol_response+sent_tokens[idx]
        return robol_response
```

+ Code + Text

```
▶ 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 r 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")
```


OUTPUT:

In general output:



A Jupyter Notebook interface showing a code cell and its output. The code cell contains Python code for a chatbot. The output cell shows the chatbot's responses to user inputs.

```
+ Code + Text
```

```
print(response(user_response))
sent_tokens.remove(user_response)

else:
    flag=False
    print("BOT: goodbye! take care")
```

27s

BOT: my name is Stark. Let's have a conversation! Also, if you want to exit any time, just type Bye!

hi

BOT: *nods*

hello

BOT: *nods*

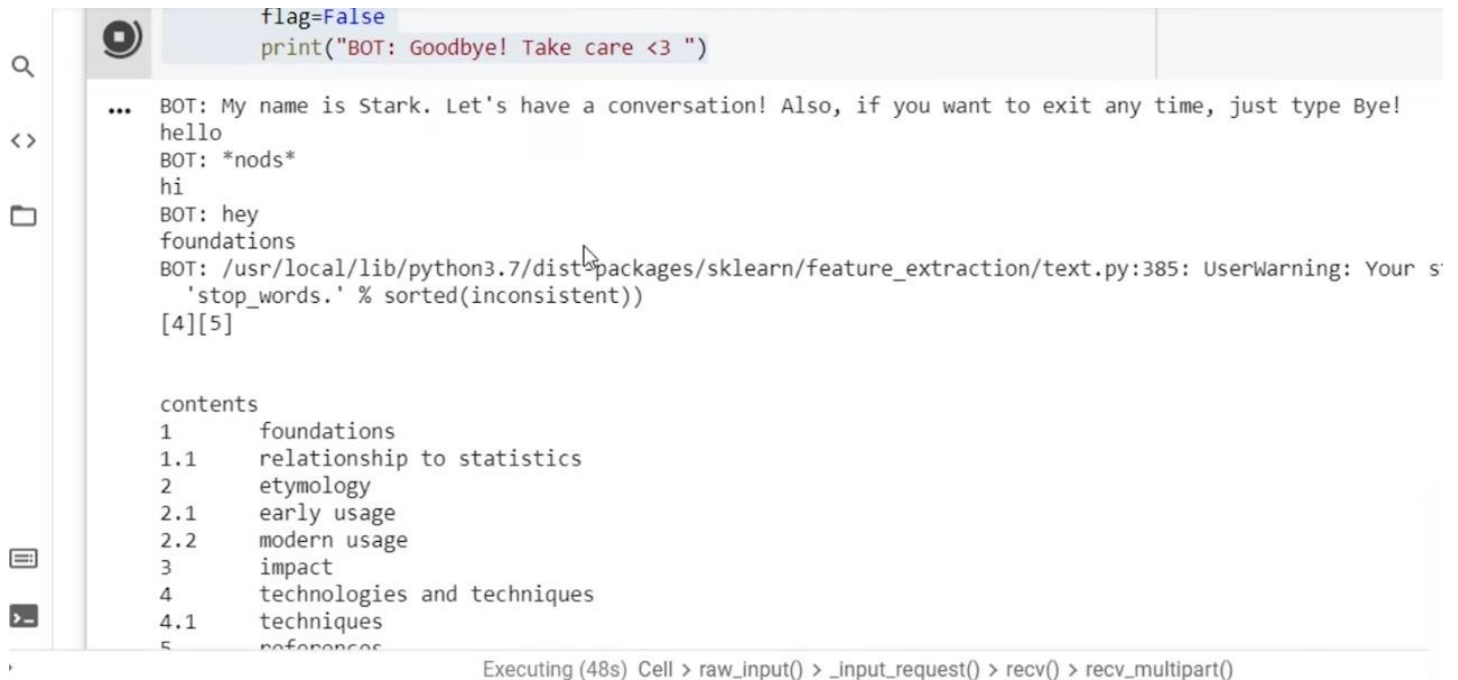
sup

BOT: hi there

bye

BOT: goodbye! take care

Referring the corpus output:



A Jupyter Notebook interface showing a code cell and its output. The code cell contains Python code for a chatbot. The output cell shows the chatbot's responses to user inputs, including a warning message and a list of contents.

```
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!

hello

BOT: *nods*

hi

BOT: hey

foundations

BOT: /usr/local/lib/python3.7/dist-packages/sklearn/feature_extraction/text.py:385: UserWarning: Your s

'stop_words.' % sorted(inconsistent))

[4][5]

contents

1 foundations

1.1 relationship to statistics

2 etymology

2.1 early usage

2.2 modern usage

3 impact

4 technologies and techniques

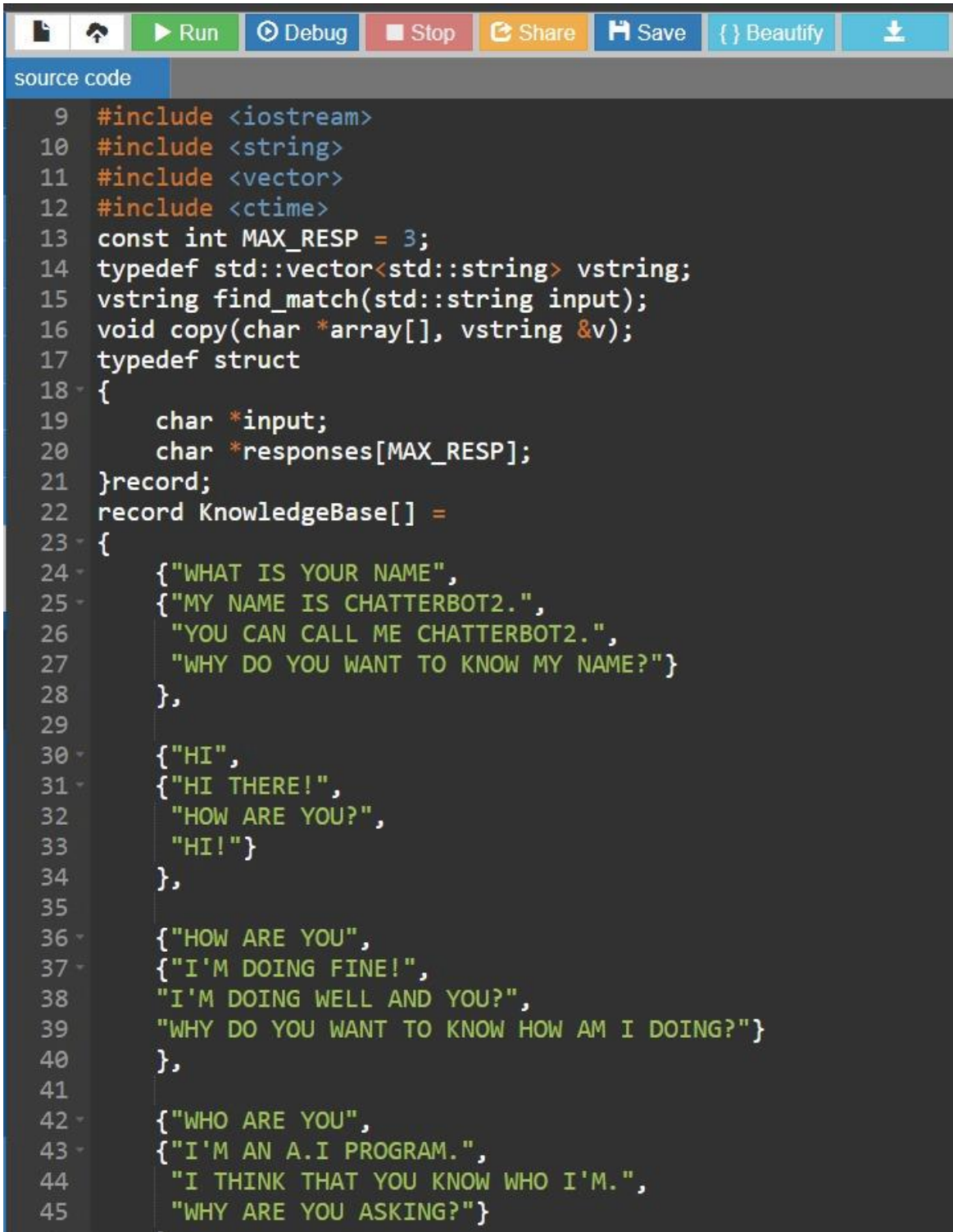
4.1 techniques

5 references

Executing (48s) Cell > raw_input() > _input_request() > recv() > recv_multipart()

RULE/KNOWLEDGE – BASED CHATBOT

CODE:



```
9  #include <iostream>
10 #include <string>
11 #include <vector>
12 #include <ctime>
13 const int MAX_RESP = 3;
14 typedef std::vector<std::string> vstring;
15 vstring find_match(std::string input);
16 void copy(char *array[], vstring &v);
17 typedef struct
18 {
19     char *input;
20     char *responses[MAX_RESP];
21 }record;
22 record KnowledgeBase[] =
23 {
24     {"WHAT IS YOUR NAME",
25     {"MY NAME IS CHATTERBOT2.",
26      "YOU CAN CALL ME CHATTERBOT2.",
27      "WHY DO YOU WANT TO KNOW MY NAME?"}
28     },
29
30     {"HI",
31     {"HI THERE!",
32      "HOW ARE YOU?",
33      "HI!"}
34     },
35
36     {"HOW ARE YOU",
37     {"I'M DOING FINE!",
38      "I'M DOING WELL AND YOU?",
39      "WHY DO YOU WANT TO KNOW HOW AM I DOING?"}
40     },
41
42     {"WHO ARE YOU",
43     {"I'M AN A.I PROGRAM.",
44      "I THINK THAT YOU KNOW WHO I'M.",
45      "WHY ARE YOU ASKING?"}
```

source code

```

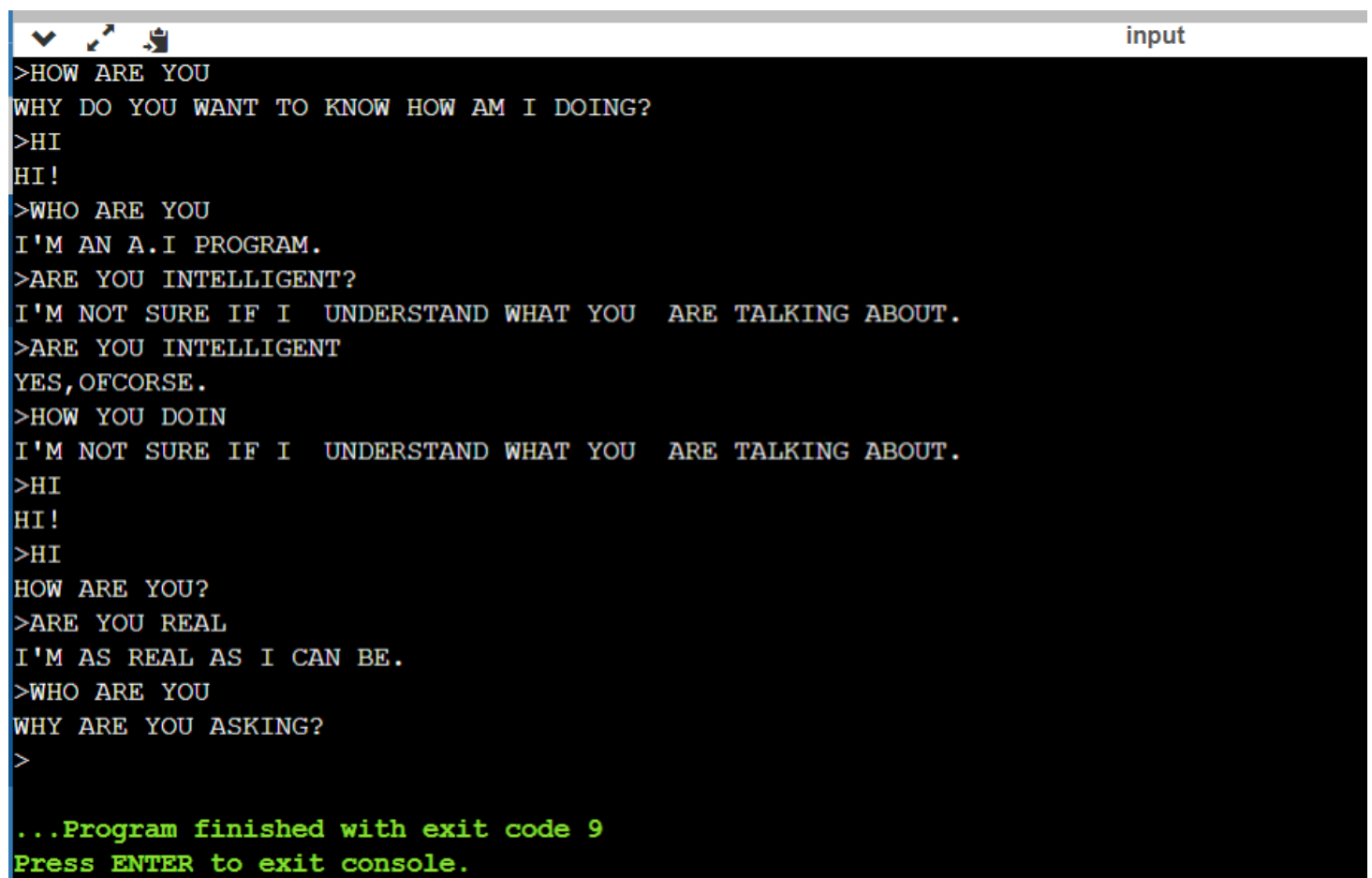
46     },
47
48     {"ARE YOU INTELLIGENT",
49     {"YES, OF COURSE.",
50     "WHAT DO YOU THINK?",
51     "ACTUALLY, I'M VERY INTELLIGENT!"}
52     },
53
54     {"ARE YOU REAL",
55     {"DOES THAT QUESTION REALLY MATTER TO YOU?",
56     "WHAT DO YOU MEAN BY THAT?",
57     "I'M AS REAL AS I CAN BE."}
58     }
59 };
60 size_t nKnowledgeBaseSize = sizeof(KnowledgeBase)/sizeof(KnowledgeBase[0]);
61 int main()
62 {
63     srand((unsigned) time(NULL));
64     std::string sInput = "";
65     std::string sResponse = "";
66     while(1)
67     {
68         std::cout << ">";
69         std::getline(std::cin, sInput);
70         vstring responses = find_match(sInput);
71         if(sInput == "BYE")
72         {
73             std::cout << "IT WAS NICE TALKING TO YOU USER, SEE YOU NEXT TIME!" << std::endl;
74             break;
75         }
76         else if(responses.size() == 0)
77         {
78             std::cout << "I'M NOT SURE IF I UNDERSTAND WHAT YOU ARE TALKING ABOUT."
79             << std::endl;
80         }
81         else
82         {
83             int nSelection = rand() % MAX_RESP;
84             sResponse = responses[nSelection]; std::cout << sResponse << std::endl;
85         }
86     }
87     return 0;
88 }
89 // make a search for the user's input
90 // inside the database of the program
91 vstring find_match(std::string input)
92 {
93     vstring result;
94     for(int i = 0; i < nKnowledgeBaseSize; ++i)
95     {
96         if(std::string(KnowledgeBase[i].input) == input)
97         {
98             copy(KnowledgeBase[i].responses, result);
99             return result;
100         }
101     }
102     return result;
103 }
104 void copy(char *array[], vstring &v)
105 {
106     for(int i = 0; i < MAX_RESP; ++i)
107     {
108         v.push_back(array[i]);
109     }
110 }

```

The program can understand some sentences like "**what is your name**", "**are you intelligent**", etc. And also, he can choose an appropriate response from his list of responses for this given sentence and just display it on the screen.

We've also added a couple of new techniques to these new programs: when the program is unable to find a matching keyword for the current user input, it simply answers by saying that it doesn't understand which is quite human like.

OUTPUT:



```
>HOW ARE YOU
WHY DO YOU WANT TO KNOW HOW AM I DOING?
>HI
HI!
>WHO ARE YOU
I'M AN A.I PROGRAM.
>ARE YOU INTELLIGENT?
I'M NOT SURE IF I UNDERSTAND WHAT YOU ARE TALKING ABOUT.
>ARE YOU INTELLIGENT
YES,OFCORSE.
>HOW YOU DOIN
I'M NOT SURE IF I UNDERSTAND WHAT YOU ARE TALKING ABOUT.
>HI
HI!
>HI
HOW ARE YOU?
>ARE YOU REAL
I'M AS REAL AS I CAN BE.
>WHO ARE YOU
WHY ARE YOU ASKING?
>

...Program finished with exit code 9
Press ENTER to exit console.
```

References

[1][IRJET-V7I51160.pdf](#)

[2][How do Chatbots work? A Guide to the Chatbot Architecture](#)
[\(marutitech.com\)](#)

[3][\(PDF\) A Smart Chatbot Architecture based NLP and Machine Learning for](#)
[Health Care Assistance \(researchgate.net\)](#)

[4][\(PDF\) Chatbot for university related FAQs \(researchgate.net\)](#)

Thank you