



# CHATBOT USING STORE AND RETRIEVAL:A SELF LEARNING APPROACH

Aniruddha Prabhu B.P

Assistant Professor

Department of Computer Science and Engineering,  
Siddaganga Institute of Technology, Tumkur  
Karnataka, India

Ritu Raj

Department of Computer Science and Engineering,  
Siddaganga Institute of Technology, Tumkur  
Karnataka, India

**Abstract**— A chatbot is an agent that can set up a natural conversation between a machine and the human. Often they are designed to give relevant information about a particular domain on which it is designed for or do a general talk or even can be used for entertainment purposes. It is designed to be a virtual assistant for various purposes. Used in business, educational, medical, agriculture and many other areas. All these are accomplished by the use of (Artificial Intelligence) AI. There are categories of chatbot which focus on general talk, while others which has most but not of specific requirement to us. To counteract this problem we designed a chatbot which would be the domain based on our requirement and also one can store our own information in it, in a way creating a dataset for the chatbot itself. Also it can do a bit random talk.

**Keywords**— *Natural Language Processing (NLP), NLTK, regular expression matching*

## I. INTRODUCTION

Today's era is a technical one and with the boom of the Internet it expanded at a fast rate. Many things are available at the Internet and with a click at the door step. With so many customers based and educational things available in it, it also gave rise to a virtual assistance to whom we can interact with. As important in business and so in educational field as well. So a large no of chatbots came into picture to interact and fulfill our needs. Chatbots made by humans interact with us learn from it and next time produce an appropriate result. My project work is based on this very concept of chatbot making. It uses python as a programming language as it is one of the most efficient when it comes for machine learning. It also uses Natural Language Processing(NLP). It is a domain based for now as I am using it for general knowledge question answer process. The domain data about general knowledge is stored in excel file format. It can also do some basic talk like other chatbot such as telling time, thanking you for wishing it happy birthday, replying your how are you or who made you questions. It also performs basic mathematical calculations. It goes for storing your questions and answer it do not know, store it in its own dataset and replying you next time when

asked. This question answer is also stored in a separate excel file. It applies natural language processing while searching for the stored questions. Storing in excel file make it very beneficial for easy retrieval purpose using simple row extraction from excel file.

## II. LITERATURE REVIEW

### A. Eliza

Considered to be the first chatterbot in the history of computer science. Created by Joseph Weizenbaum, a German computer scientist at Massachusetts Institute of Technology in 1966.

Now lets see how it works

Here is an example from the paper "Eliza computer program for the study of natural language communication between man and machine"

Taking up an example of the sentence:-

"It seems that you hate me" entered by the user.

Dividing into components

1->It seems that

2-> you

3->hate

4->me

Understanding the two parts (2) and (4) it translate it into I and you respectively.

And attach a stock phrase like "what makes you feel"

The sentence after farming becomes

"What makes you think I (3) you. [3]

This particular ChatBot was not having a storing capability to store the previous interaction from the user and then go for processing.



## B. ALICE

ALICE stands for Artificial Linguistic Internet Computer Entity. It stores knowledge of English conversation in AIML files.

AIML consist of AIML an object that has topics and categories.

Topics are optional having name and set of categories related to topics. Categories are the unit of knowledge in AIML. It consist of rule for matching an input and converting it into output. Its structure comprise of template which is basis of ALICE answer. [4]

With more and more development made in the chat bot we now have chat bot that can be combined with sensing emotion like given some words by the user and the chat bot can be able to predict response on word content and emotional level [2].

ChatBot can be made for any of the purpose but they are basically of two types. One is retrieval and the other is the generative type.

Retrieval type of chatbots are primarily based on the predefined responses that are given to the user when they ask some query. Here the asked query is matched with the predefined set of responses. This tends to pose no grammatical mistake, just match and display.

The generative type is the one that is complex and it utilizes ongoing conversation with the user in used to frame the correct response. Here the chances of accuracy is not always hundred percent. They may be domain specific. [1]

This chatbot uses both the above types in combination to satisfy its need.

## III. EXPLANATION OF PROCESS FLOW FOR CHATBOT

Now let us see the basic implementation stages involved in the making of the making of this ChatBot. First of all let us see the flow chat involving various stages to it as shown in the Fig. 1. It starts from accepting the input from the user in the form of text. Then it goes on for processing each word that can be matched for a particular action to be taken like storing, deleting, going for random talk ,asking time, key word matching for domains and if not going for or storing it in the store file(which is the name used in the whole context for the data file,in this case is the excel sheet,that would store the interaction which the ChatBot has no previous stored knowledge in the form of question and answer).

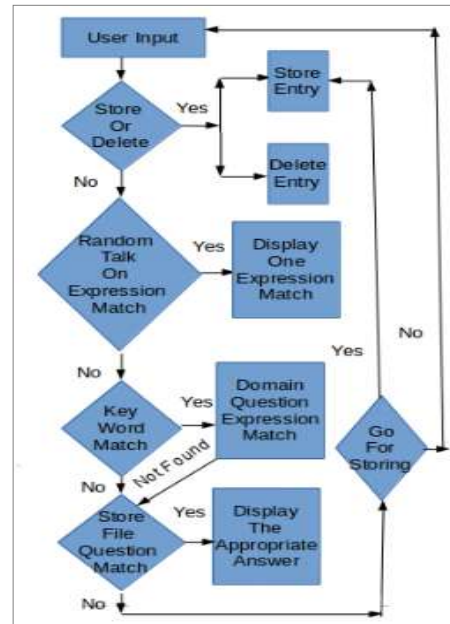


Fig. 1. Flowchart for working of Chatbot

### A. Stage 1

Random talk would be hi,hello,who made you direct questions which are frequently asked so it would directly give the required previously stored answer.

```
l=["hi", "hello", "hey", "how are you", "hey you"]
#=["what can you do", "whats your domain", "whats you
n=["your creator", "who build you", "who created you"]
o=["what is todays date", "what date is today", "todas
p=["what time is now", "current time", "time", "what i
```

Fig 2. Random talk questions

If the user wishes all the best for (-----) or best of luck for (-----), in this case python regular expression matching is used and the word (-----) is extracted and a new sentence in reply is framed and displayed.

```
def wishing(k):
    m1=re.match(r'good (.*)',k,re.M|re.I)
    m2=re.match(r'happy (.*)',k,re.M|re.I)
    m3=re.match(r'all the best for (.*)',k,re.M|re.I)
    m4=re.match(r'you are (.*)',k,re.M|re.I)
```

Fig. 3. Wishing using regular expression matching

Current time and date can also be displayed when asked.

### B. Stage 2

Now the domain for which it is made for come into picture. Here also python regular expression matching is used. First keywords are matched which will direct it to from which file data is to be fetched. (in this project I used country-capital and a book-author excel files so the keywords were country, capital, currency, language for country-capital file and book,author for book-author file).



```
#1=re.match('capital of (.*)',k,re.M|re.I)
#2=re.match('which is the capital of (.*)',k,re.M|re.I)
#3=re.match('what is the capital of (.*)',k,re.M|re.I)
#4=re.match('capital of (.*) is',k,re.M|re.I)
#5=re.match('which state capital is (.*)',k,re.M|re.I)
#6=re.match('currency of (.*) is',k,re.M|re.I)
#7=re.match('currency of (.*)',k,re.M|re.I)
#8=re.match('(.*) is the official language of',k,re.M|re.I)
```

Fig. 4. Already framed question

There are different questions already framed from which using regular expression matching we get to know what is being asked and for that what is to be fetched first Example what is the capital of (-----) here it is asked about the capital and the country is the column which we need to fetch first.

```
def countryc(s):
    with open('city.csv') as csvfile:
        reader = csv.DictReader(csvfile)
        k=0
        for row in reader:
            if s.lower()==row["Currency"].lower():
                k+=1
                return k,row["Country"]
    return k,""
def country(s):
    with open('city.csv') as csvfile:
        reader = csv.DictReader(csvfile)
        k=0
        for row in reader:
            if s.lower()==row["Capital"].lower():
                k+=1
                return k,row["Country"]
    return k,""
```

Fig. 5. Fetching from the files

Also a simple calculator is being implemented which does basic calculation. It searches for the keyword /,\*,-,+.

### C. Stage 3

The raw text has been entered by the user it is being sent through the domain section. In this if the text exactly matches the store file question then the answer to the particular question is being displayed.

If not then all the question tag words like which,what,why and also words like is,the,an,of is removed from the text entered by the user and also in each of the question of the store file. This is done using word tagging technique of NLTK.

```
def domain(a):
    a=a.strip()
    k=0
    with open('some.csv') as csvfile:
        reader = csv.DictReader(csvfile)
        for row in reader:
            if a.lower()==row["Question"]:
                print row["Answer"]
                k+=1
                return k
    m=domain1(a.lower())
    return m
```

Fig. 6. Matching the text with the stored questions

`nltk.tag.pos_tag(['what'])` is used to find part of speech of speech for a word. In this case the word is 'what'.

```
def tokenn(e):
    tokens=nltk.word_tokenize(e)
    tagged=nltk.pos_tag(tokens)
    words=[w for w,pos in tagged if (pos=='VGB' or po
```

Fig. 7. Tagging and listing words

Now the no of words that are matched in both the question and the text is counted. This is done for every question and a frequency list is being created.

The largest no among those frequency list is selected and the corresponding answer from the store file is displayed.

In case there are two max frequency is there then chatbot would ask for more specific text to enter.

```
with open('some.csv') as csvfile:
    reader=csv.DictReader(csvfile)
    for row in reader:
        del r[:]
        row["Question"]=row["Question"].lower()
        row["Question"]=removeperiod(row["Question"])
        x=row["Question"].split()
        t=tokenn(row["Question"])
        for v1 in x:
            if v1 not in t:
                r.append(v1)
        k=match(q,r)
        n.append(k)
        n1.append(k)
```

Fig. 8. Creating a frequency of words matched

### D. Storing and deleting stage

If none of the above matches then it goes for storing it in store file in an interactive way. If we don't want to store simply give a n or no when asked. In the figure 9 shown below there is a removeperiod function which actually removes all the . , ? like symbols from the text and then store it.

```
a=a.lower()
a=removeperiod(a)
b=raw_input()
d.append(a)
d.append(b)
with open('some.csv', 'a') as f:
    writer = csv.writer(f)
    writer.writerow(d)
    del d[:]
    return
```

Fig. 9. Storing the data in the store file

On the basis of such considerations, the algorithm uses a different color image multiplied by the weighting coefficients of different ways to solve the visual distortion, and by embedding the watermark, wavelet coefficients of many ways, enhance the robustness of the watermark.

We can even delete an entry from the store file using some steps.

This would clearly help us to get away with all the data or information we don't want to store and or stored incorrectly

First copy all the entries except leaving the entry which we want to delete in another file.

Delete the previous file



Rename the new file to the previous file name.

```

if k==1:
    d=['Question','Answer']
    with open('some1.csv','w') as f:
        writer = csv.writer(f)
        writer.writerow(d)
    with open('some.csv') as csvfile:
        reader = csv.DictReader(csvfile)
        for row in reader:
            del e[:]
            if row['Question']!=a.lower():
                with open('some1.csv','a') as f:
                    writer = csv.writer(f)
                    e.append(row['Question'])
                    e.append(row['Answer'])
                    writer.writerow(e)
            else:
                continue
    os.remove('some.csv')
    os.rename('some1.csv','some.csv')
    
```

Fig. 10. For deleting entries from the store file

Deletion and storing can be done at beginning stage by just giving store and delete command.

#### IV. EXPERIMENTAL RESULTS

Now moving on to see the results of each and every stages. Starting from the random talk to the storing in the domain all are based on the actual running of the ChatBot on the system.

All the results are taken from the actual running of the proposed chatbot on the terminal in the linux operating system.

```

=====
CHAT>
hi
welcome to a nice talk
=====
CHAT>
hello
hey you
=====
CHAT>
what is todays date
12/1/2018
=====
CHAT>
time
14:1:21
=====
    
```

Fig. 11. Random Talk

In Fig. 11. user enters hi,hello,asks the time or the date and the chatbot responds by giving appropriate answers. In case of time and date it gives the current time and date. Here there is framed responses for the questions that we are going to put. A small regular expression matching task is done to know what the response user want to get.

```

=====
CHAT>
happy christmas
wish you too a happy christmas
=====
CHAT>
you are good
Thanks for telling me good
=====
    
```

Fig. 12. Wishing the chat bot

In Fig 12. it extracts the word after happy and present in proper format as a reply.

```

=====
CHAT>
which is the capital of Australia?
Canberra
=====
CHAT>
currency of Australia
Australian Dollar
=====
CHAT>
official language of australia
English
=====
CHAT>
who is the author of a tale of two cities
Charles Dickens
=====
CHAT>
which is the book written by vikram neth
A Suitable Boy
=====
    
```

Fig. 13. Chatbot answering the domain questions

Here in Fig. 13. Chatbot answers the questions for the domain it was made. As from the figure user asks about the currency language of Australia and also book's author, also name of the author who wrote a particular book.

```

=====
CHAT>
calculate 2+2/2
[3]
=====
CHAT>
321456/8947-89647/2145
[-6]
=====
    
```

Fig. 14. A simple calculation

The ChatBot performs basic calculation such as +,-,\*/ and also follows the basic rule for calculation as shown in Fig. 14.

```

=====
CHAT>
color of the sky
do you like to store it
=====
y
what you like to store about
=====
blue
=====
CHAT>
what is the color of the sky?
blue
=====
    
```

Fig. 15. Out of domain question storing by chatbot.





In Fig. 15. When asked out of domain question it asks the user to store it and also what to store about. Then when asked about it later, the Chatbot replies appropriately.

Here both the question 'color of the sky' and 'blue' is stored in question and answer format respectively in the store file. When asked second time the Chatbot first matches the exact words if not present then it goes for applying Natural Language Processing, calculating the frequency of each word matching and getting the maximum number which in this case maximum frequency is for 'color of the sky'. The answer in the store file for that corresponding question is 'blue'.



Fig. 16. Storing and deletion done by Chatbot

In Fig. 16. store keyword is used which would directly store the given context in the store file.

First topic name is asked and then what we want to store about.

Here 'apex' is the word and 'nothing' is the text we want to store about word 'apex'

Now we want to see if it is stored so we enter 'apex' and the answer is 'nothing', so it is properly stored in the store file.

Now we want to delete the word 'apex' and the required information about it, so we enter delete keyword and entered 'apex'.

We want to check if it is removed from the store file or not so we go for again entering 'apex', Chatbot would ask whether we want to store it or not which means it is not present in the store file anymore.

## V. ADVANTAGES OF THIS APPROACH

It is domain based which is can be created by our own need and can be easily extended

Its ability to store information can further add extension to it which is very helpful when we want something out of domain.

With the ability to store out of domain information it can dynamically increase its dataset through interaction with it.

It acts as a virtual assistance.

It can be used for personal purposes since it can be easily made according to our required domain.

We can even delete from the store data which we stored during our interaction.

## VI. CONCLUSION AND FUTURE IMPLEMENTATION

This ChatBot was made on store and retrieve procedure. The main idea is to use the dynamic data which we are getting and then processing it and displaying the required output. The Chatbot can be further extended to a larger scale by including a large no of domains ranging from information stored about various cities to displaying about diseases spread by viruses. From awards won by people to movie reviews. It can be easily extended by just including the rows and column based information about that particular domain.

## VII. ACKNOWLEDGMENT

I give my special thanks to Aniruddha Sir for his constant guidance and support. It has been a wonderful experience working for him. The whole idea construction has been the result of our collaboration. The constant support for research and project development in our college also makes us do a lot of productive work in the field of technology. So it was a all around support for me to make this happen.

## VIII. REFERENCES

- [1] Z Babar , A Lapouchnian and E Yu on "Chatbot Design - Reasoning about design options using i\* and process architecture"
- [2] H WEI, Y Zhao and J Ke on "Building Chatbot with Emotions"
- [3] J. Weizenbaum "Elizaa computer program for the study of natural language communication between man and machine", Communications
- [4] B AbuShawar and E Atwell in the paper "Automatic Extraction of Chatbot Training Data from Natural Language Corpora"