IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Vol. 10, Issue 5, May 2021

DOI 10.17148/IJARCCE.2021.10532

AN NLP CHATBOT ON HOME REMEDIES FOR COMMON DISEASES

K N Prathik[1], Dr Rajashree Shettar[2]

Department of Computer Science and Engineering, RV College of Engineering® Bengaluru, India[1,2]

Abstract: Conversational systems have made considerable strides in recent years as a form of human-computer interaction. Human-computer interaction has paved the way for large-scale natural language processing techniques. A chatbot in medical field [10] does instant research on people's problems and prescribes medicine and assistance without any delay [9]. This system proposes a medical assistant chatbot that uses NLP concepts such as cosine similarity and TF-IDF approaches to replay to users, which will be pocket friendly, fast, accurate and a one place halt for a user to avail home remedies and easy cures for common diseases.

Index Terms: Natural Language Processing (NLP), Natural Language Toolkit (NLTK), Term Frequency-Inverse Document Frequency (TF-IDF), Cosine Similarity.

I. INTRODUCTION

Chatbots are commonly used in a variety of fields, including industry, education, healthcare, and many others. A chatbot in medical field does instant research on people's problems and prescribes medicine and assistance without any delay. The proposed medical assistant chatbot application uses the natural processing engine and the basic design of a chatbots in order to achieve the expected level of accuracy. The steps involved in NLP chatbot on home remedies for common diseases involves text pre-processing, Text must be transformed into a logical sequence (or vector) of numbers after the initial pre-processing step. Querying the database to respond user request. Text pre-processing is tedious job because it involves changing the entire string to uppercase or lowercase, so that the algorithm would not confuse similar terms in different situations and Tokenization, Stemming, Lemmatization, Excluding Noise, Excluding Stop Words. The TF-IDF solution and cosine similarity are used to derive the meaningful vector of numbers from the pre-processed data.

II. SYSTEM ARCHITECTURE

The main objective of this project is to build a domain specific data driven chatbot. The scheme of the chatbot design is as shown in figure 1. The solution suggested using the rows and columns of very dataset that is to be queried to train the model to convert user utterances into queries that can be run on the database. The data hence retrieved is shaped to resemble a human conversation and displayed to the user.

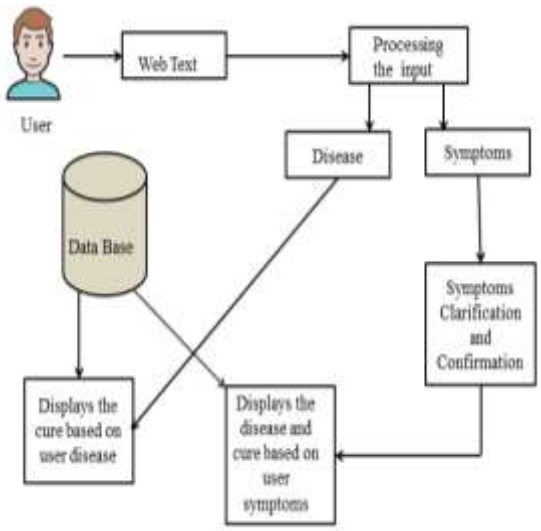

Figure.1 System Architecture

IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Vol. 10, Issue 5, May 2021

DOI 10.17148/IJARCCE.2021.10532

The system architecture is shown in figure 1. Proposed architecture makes the chatbot a lot more versatile and flexible. The system can be widely broken down into two parts.

- User Interface
- Chatbot model

A. User Interface

This section of the application deals with the frontend GUI, i.e. what the user will be entering into the text space and what the user will be seeing when the chatbot returns the retrieved data in a conversation template [7]-[6]. In order render HTML pages from the python code, flask and rendor_template are imported from the flask library. Fig 2 shows pseudo code for connection to interface

```
from flask import Flask,
redirect, url_for, request,
render_template

app = Flask(__name__)

from flask_cors import CORS,
cross_origin

cors = CORS(app,
resources={r"/medi":
{"origins":
"http://localhost:5000"}})

@cross_origin(origin =
'localhost', headers =
['Content- Type',
'Authorization'])
```

Figure.2 Pseudo code for connection to interface

In order to make the Cross-origin AJAX possible the CORS (Cross-Origin Resource Sharing) is imported from the flask library. The Resource specific CORS and the Route Specific CORS is used via decorator to allow CORS on a given route.

B. Chatbot Model

This section contains the core functionality of the project, the system that converts utterances to database queries and moulds the retrieved data to follow a conversational template.

III. RELATED WORK

Ticketing Chatbot Service using Serverless NLP Technology **[1]** focuses on the Chatbot as a forwarding agent capable of classifying user meaning in a conversation. Natural language processing (NLP) was used by the chatbot to evaluate the query and retrieve some keyword information. Two crucial NLP processes are morphological examination and part of speech (POS) marking. POS assists in interpreting the meaning of chat text using a set of rules. The rule base is specific to a language and is designed to capture all keywords found in chat text.

NLP chatbot for Discharge Summaries **[2]** propose a computer-human Interaction framework that can respond to patient inquiries and improve the quality of medical services provided to an individual. In this case, Natural Language Processing and Big Data were used. Natural Language Processing To process the data in sklearn, libraries like ntlk and nlp are used. The resulting abstract "topics" are discovered by statistical subject analysis.

The Combination of Natural Language Processing and Entity Extraction for Academic Chatbot. **[3]** The provided chatbot employs the NLP (Natural Language Processing) algorithm and entity extraction system, allowing it to react accordingly and in accordance with student requests. The NLP algorithm is a text mining feature that performs a particular type of linguistic analysis to help machines "interpret" text. Entity extraction is a form of information recognition that entails recognizing and categorising critical components of text into predetermined groups.

IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Vol. 10, Issue 5, May 2021

DOI 10.17148/IJARCCE.2021.10532

Human-Computer text conversation through NLP in Tamil using Intent Recognition **[4]** focuses on creating a Chatbot that uses Natural Language Processing (NLP) and the Dialog Flow API to respond parent's questions in their primary tongue (Tamil), facilitating parents in keeping a record of their students' college performance.

Towards Automating the Synthesis of Chatbots for Conversational Model Query **[5].** In this method, the Xatkit framework chatbot development platform interprets the developed chatbot system and communicates with an EMF (Eclipse Modeling Framework) backend.

IV. METHODOLOGY

This paper, proposes the use of Tokenization, Stemming and Lemmatization for text pre-processing and TF-IDF approach and cosine similarity to derive the meaningful vector of numbers from the pre-processed data.

A. Pre-processing text with NLTK

Textual information's only flaw is that all this is written in strings. However, the algorithms, on the other hand, require some kind of quantitative vector to complete the task. The NLTK data kit includes a pre-trained Punkt tokenizer for English. The algorithm does not distinguish between the same words in different situations, as a result, simple text pre-processing entails changing the text's case to uppercase or lowercase. Tokenization is a process of collection of tokens from regular text strings, which includes eliminating noise and stop terms**.** Some particularly common terms that tend to be of little use in assisting in the selection of documents that meet a user's needs are occasionally removed entirely from the vocabulary. Lemmatization is the act of eliminating inflectional endings from a word and returning it to its base or dictionary form, which is defined as lemma. For example, the term "run" is a base form for terms like "running" or "ran," and the terms "better" and "nice" will be in the same lemma, thus they are considered synonymous**.** Fig 3 shows Pseudo code for Lemmatization.

```
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(tex
t.lower().translate(remove_punct
_dict)))
```

Figure.3 Pseudo code for Lemmatization

B. Derive the meaningful vector from the pre-processed data

The proposed system uses the TF-IDF approach and cosine similarity to derive the meaningful vector of numbers from the pre-processed data.

TF-IDF Approach- This method includes rescaling the frequency of term according to how frequently they occur throughout all records, penalising recurrent terms like "the" that usually show in all documents [8]. Term Frequency-Inverse Document Frequency is a scoring method, where:

IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Vol. 10, Issue 5, May 2021

DOI 10.17148/IJARCCE.2021.10532

Term Frequency: is a score dependent on how much a term appears in the current document. Fig 4 shows Term Frequency Formula.

TF – (Number of times term t appears in a document) / (Number of terms in the document)

$$tf_{i,j} = \frac{n_{i,j}}{\sum_k n_{i,j}}$$

Figure.4 Term Frequency Formula

Inverse Document Frequency: is a metric for determining how uncommon a word is through documents. Inverse Document Frequency Formula is shown in fig 5.

IDF = 1+log(N/n), where, N is the number of documents and n is the number of documents a term t has appeared in.

Figure.5 Inverse Document Frequency Formula

The TF-IDF weight is a popular choice for information retrieval and text mining. This measure is used to determine the significance of a term in a list or corpus of documents.

Cosine Similarity

Any pair of vector's Cosine similarity can then be determined by dividing their dot product by the product of their norms [8]. The cosine of the angle created by the two vectors is obtained in this way. Cosine similarity is a measure of how similar two non-zero vectors are. This formula is being used to evaluate the degree of similarity between two documents d1 and d2. Fig 6 is Cosine Similarity Formula.

Cosine Similarity (d1, d2) = Dot product(d1, d2) / ||d1|| * ||d2||

Figure.6 Cosine Similarity Formula

C. Dialogue Management

The database needs to be processed. Pre-processing will include loading of the database which is in csv format into data frames and cleaning the data in the data frames. The CSV needs to be loaded into columns in order to make it quarriable by various persistence APIs that in turn run queries.

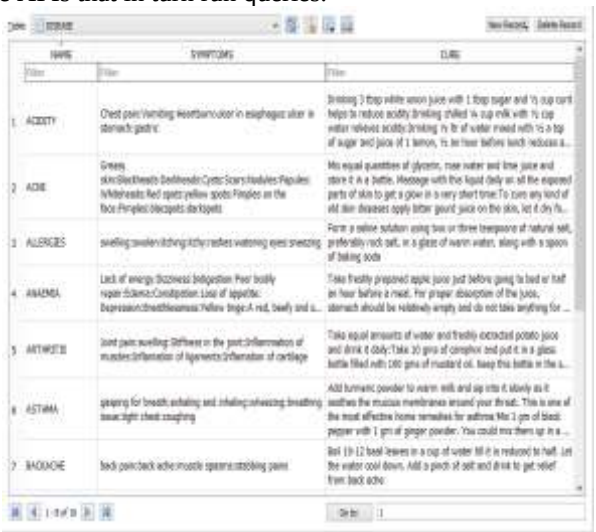

Figure.7 Backend Table

IJARCCE
ISSN (Online) 2278-1021
ISSN (Print) 2319-5940
International Journal of Advanced Research in Computer and Communication Engineering
Vol. 10, Issue 5, May 2021
DOI 10.17148/IJARCCE.2021.10532

Intent text and query text might include the entities which might be referring to the verticals of the database. So this will be implemented by extracting different forms like verb, adverb and other forms of the column and storing it in a list and reading through the text to check if any of the tokens matches any of the columns' variations. After system recognize the columns, relating those columns to relevant token to extract the parameter is to be done. Fig 7 shows BackendTable.

D. Dialogue Control

Dialogue control will form a request using sentences with placeholders and take the response passed down from database server and form a relevant response that follows the conversational template, which will be displayed in the user interface. Fig 8 shows Pseudo code for Link to the Backend.

```
$("#d1").on('click', function(event) {

        $("#dcw").append("<div
class=bubble-
right>"+$("#nm").val()+"</div>")

        });

$("#nm").on("keypress", function(e){
        if(e.which == 13)
{
$("#dcw").append("<div class=bubble-
right>"+$("#nm").val()+ "</div>");}});
```

Figure.8 Pseudo code for Link to the Backend

V. RESULT AND ANALYSIS

The chatbot is designed for providing appropriate and relevant cure for the common diseases through traditional medicines and home remedies. It undergoes accurate analysis of the symptoms and provides the random cure for the disease. This section provides the final outcome of the project through various snapshots of the output.

Figure.9 Home page

The fig.9 shows the Home page of the chatbot system which provides the user a chat window to converse with "Homeo" to get the cure for the relevant disease or symptoms. It is also provided with the "Clear Chat" button to clear the old chat and to start a new conversation.

Figure.10 Greeting messages with bot

IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Vol. 10, Issue 5, May 2021

DOI 10.17148/IJARCCE.2021.10532

The fig.10 shows that the bot is able to respond to the greetings and can continue conversation with the user for such requests.

Figure.11 Cure for Input Disease

The fig.11 shows the chat messages where the user asks the bot on a particular disease and the bot responses with the appropriate cure for the disease

Figure.12 Cure for Input Symptoms

The fig.12 shows the chat messages where the user asks the bot with the disease symptoms and the bot analyses the symptoms and responds with the disease name and the relevant cure for that disease.

Table 1 shows the 14 scenarios that were checked in the chatbot. There are two conditions in the 14 cases that do not fulfil the output expectation.

Table.1. Intent Testing Scenario

Test Scenario	Input
Do a greeting messages like hi, hai, hello, thank you, bye, whats up	User say "hi"
	User says synonym of hi(hai, hello, ...whats up)
	User says synonym with type(hai or helloee)
	User says "bye"
Do a disease name typing in chat	User says "Fever"
	User says "I have fever"
	User says "I have fever and cold"
	User says "I have fever, cold, cough"
Do a symptoms typing in chat	User says "I have high body temperature"
	User says "I have high body temperature, Head ache, body pain"
	User says "I have high body temperature and body pain"
	User says "I have high temp"
Do a click in cureit button	User clicks cureit button
Do a click in clear button	User clicks clear button

In chatbot checking, valid responses (relevant answers) must be counted as 1 and invalid responses (non-relevant answers) must be counted as 0 [1]. Tabel 2 shows the notation of relevant testing.

Table.2. Testing Result of Scenario

	Relevant	Non-Relevant
Retrieved	12	2
Non Retrieved	0	0

IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Vol. 10, Issue 5, May 2021

DOI 10.17148/IJARCCE.2021.10532

We can infer precision, recall calculation from Table 2. Since the chat always replied, recall is 100 % and the precision rate of intent testing is 85.71%. Precision and recall have a harmonic mean of 92.3%, which is calculated by dividing precision times recall by precision plus recall twice.

VI. CONCLUSION

The designed chatbot can detect a disease and provide the suitable cure and we have been successful in developing a chatbot that performs excellently for non-ambiguous queries and would be useful for personnel who have no knowledge of any database to access the database and query the resource they want. Functionalities of the chatbot can be further enhanced by adding aggregators or analysis options on the data. Chatbot can also act as a common interface between databases and outside application which use it. The chatbot aims to be a stand-alone application, which institutions or any organization can plug to their own data resources and utilize with minimal set up effort.

REFERENCES

[1]. Eko Handoyo, and M. Arfan. "Ticketing Chatbot Service using Serverless NLP Technology." In 2018 5th International Conference on Information Technology, Computer and Electrical Engineering (ICITACEE), Semarang, Indonesia, pp. 325-330. IEEE, 2018.
[2]. Harsh Lal, Priyanshu Lal. "NLP chatbot for Discharge Summaries." In 2019 2nd International Conference on Intelligent Communication and Computational Techniques (ICCI), Jaipur, India, pp. 250-257. IEEE, 2019.
[3]. Lukas Tommy, chandra Kiran, Leo Riska. "The Combination of Natural Language Processing and Entity Extraction for Academic Chatbot." In 2020 IEEE International Conference on Cyber and IT Service Management(CITSM), pp. 1-6. IEEE, 2020.
[4]. Udayakumar Shanmugam, Sowjanya Mani, Sneha Shivakumar and Rajeshwari P. " Human-Computer text conversation through NLP in Tamil using Intent Recognition." In 2019 IEEE International Conference on Vision Towards Emerging Trends in Communication and Networking(VitECoN), Chennai, India, pp. 1-5. IEEE, 2019.
[5]. Sara Perez-Soler., Gwendal Daniel, Jordi Cabot. " Towards Automating the Synthesis of Chatbots for Conversational Model Query." In Springer BPMDS 2020/EMMSAD 2020, LNBIP , pp. 257-265. Switzerland, Springer, 2020.
[6]. S. du Preez, M. Lall, and S. Sinha, "An intelligent web-based voice chat bot," in EUROCON 2009, EUROCON'09. IEEE. IEEE, 2019, pp. 386–391.
[7]. Jiyou Jia, "A Computer assisted English learning Chatbot based on textual Knowledge and Reasoning (CSIEC)" in Elsevier Knowledge-Based Systems, Volume 22, issue 4, May 2009.
[8]. Papiya Mhajan, Rinku Wankhade, " Healthcare Chatbot using Natural Language Processing", International Research Journal of Engineering and Technology (IRJET) , Volume 07, issue 11, May 2020.
[9]. Rohini R T, Dr. Shanta Rangaswamy, "Virtual Medical Prediction using Artificial Intelligence (AI)" in International Research Journal of Engineering and Technology (IRJET), Volume 07, issue 5, May 2020.
[10]. Mrs. Rashmi Dharwadkar , Dr.Mrs. Neeta A. Deshpande, "A Medical ChatBot" in International Journal of Computer Trends and Technology (IJCTT), Volume 60, issue 1, May 2018.