International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 08 Issue: 04 | Apr 2021 www.irjet.net p-ISSN: 2395-0072

Multinomial Model Functioning with Unparsed Parameter to Reduce Response Time

Vikrant Kumar Singh[1], Vishal Kumar Singh[2], Dr. Ganesan R[3]

[1]*Student, Computer Science & Engineering, SRMIST Kattankulathur, Chennai, T.N, India*
[2]*Student, Computer Science & Engineering, SRMIST Kattankulathur, Chennai, T.N, India*
[3]*Asst. Prof, Dept. of Computer Science & Engineering, SRMIST Kattankulathur, Chennai, T.N, India*

Abstract - *Chatbot is a very popular application nowadays, because it responds with human interaction. This paper covers the design as well as algorithm implementation involved in Chatbot building for health care. The Chatbot was designed using some basic web development tools . Dataset was cleaned and relevant features were extracted. Tokenization as well as stemming was performed further and the output was passed as a parameter to the model. In this paper a novel technique has been proposed to feed the parameter to the Multinomial Naive Bayes model. Custom functions were generated where the model was fed with a string vector, no parsing of vector to integer vector was done, ultimately resulting in response time complexity reduction. Main emphasis was given on response time as well as accuracy of the model.*

Keywords **- Chatbot, Pattern Matching, Communication, Panda.**

I. INTRODUCTION

Ever since computers have been invented every passing day it's making our life convenient.It has always been useful to reduce time consumption. These computers work upon certain programs written by programmers. Chatbot is one kind of such program which makes it human interactive. Chatbot plays a vital role in all fields nowadays, even if it is a medical, engineering, business, defence field etc. In this digital era it has become a kind of companion to mankind. A Chatbot is designed to imitate as a human when a user gives a text or speech as input. This paper is based on text input given by the user. For example, if the user provides input as "Hello, how are you?", then the bot replies back with a message "I'm fine. How's you?" or "I'm good, how I can help you?" According to the sentences provided by the user it responds in the respective ways. The backend of the Chatbot user interface does the entire processing of the sentence. All chabots work on the basis of pattern matching, the sequence of words or sentences are recognized by the bot back end and response is generated. This paper covers the techniques & technologies used to design and execute the Chatbot. Identification of features, duplicate removals, incorporating algorithm, text extraction is discussed further in this paper.

II. LITERATURE SURVEY

Chatbot applications can be divided into four groups such as goal-based, knowledge-based, service-based and response generated-based. To understand better see *Fig 3*.Various examples of pre-existing chatbots are present around us like ChatScript, Alicebot, Mitsuku, Microsoft LUIS , Chatfuel, Cleverbot , IBM Watson, Elizabeth Bot .But all have some flaws over their advantages which somehow doesn't make them a perfect or efficient bot in a manner. Alicebot is also incapable of generating appropriate responses, no reasoning capabilities and unable to generate human-like responses (Turing test). On the other hand the shortcomings of Elizabeth bot is, it does not provide a way to partition or split the user input sentence and then combine their results [2]. According to Elizabeth bot's structure, it will be difficult to do the splitting. Watson has some major drawbacks such as it does not process structured data directly, no relational databases, higher maintenance cost, targeting towards bigger organizations and takes longer time and effort to teach Watson in order to use its full potential. Whereas LUIS integrates seamlessly with the Azure Bot Service, making it easy to create a sophisticated bot [3]. Chatbot developed for legal practices makes use of Heuristic evaluation and Cognitive Walkthrough approach, which makes use of voice services so somehow makes it time consuming and isn't a better approach or idea. While the chatbot made for Dialogue Learning with Human teaching and feedback uses Imitation learning and reinforced learning as a model training algorithm but results in higher cost due to much human interaction. Dialog Flow is known as Api.ai and it was developed by Google and is a part of Google Cloud Platform. Dialogflow recognizes the intent and context of what the user says. Then match user input to specific intents and use entities to extract relevant data from them. And finally, allow the conversational interface to provide responses. The drawback of Dialog Flow is no handheld device version, not interactive user interface and poor

documentation [4]. The chatbot for college management system turned out to be the nearest approach to our goal chatbot, since the Multinomial Naive Bayes algorithm was used and well trained models were applied there with an efficiency of around 87%. But there were chances of zero frequency error while predicting the class of data.

III. PROPOSED WORK

The proposed system used a novel technique to feed the parameter to the Multinomial Naive Bayes model. Custom functions were generated where the model was fed with a string vector, no parsing of vector to integer vector was done, ultimately emanated in response time complexity reduction. The parsing time complexity could be comprehended using (2) . Since the proposed model uses moderate features, if parsing was done the time complexity would have increased to some extent. Therefore feeding the model an unparsed parameter made an enhancement to the system. More focus was given on pre-processing and feature selection process. As a result no normal distribution and nicely processed data for model input. Zero frequency error was resolved by applying the smoothing algorithm or techniques like laplace correction, and transformation method was used to convert it into normal distribution if found continuous. The model used (1) for probability calculation.

A. Equations

$$P(A \mid B) = P(B \mid A)\, P(A) \quad (1)$$

where A and B are events and P(B) ? 0.

B. Time Complexity

$$O(S*N*M*N) \quad (2)$$

where , S is iterations, N is number of features, M is number of sample datasets.

IV. DESIGN OF CHATBOT

A Chabot pretends to be a human giving response as per the user input, i.e the user feels like on the other side a human is talking. Answers to the queries asked by the user are given. But how does it communicate to the user? This phenomena can be elaborated using a use case diagram, as depicted in Fig 1. During initial steps of the Chatbot design, some technicalities were kept in mind:-

A. *OS selection*

LINUX OS was used in order to perform all programming due its high speed kernel design and in addition, it provides a user friendly terminal.

B. Selection of Software

Visual Studio Code IDE (provided by Microsoft) was used for programming in python. Due to it's built-in plugin support and smartsense or intellisense for completing and detecting code.

C. Selection of Language

Though multiple languages are available for programming like - R, Java etc but python was opted because of its ease of use, built-in supports such as NLTK for text manipulation ,easy file manipulation supports (.csv, .txt, .bak, json), libraries like panda for dataset cleaning are provided.

D. Creating a Chatbot

Chatbot was created using python language to make it easy to comprehend.

E. Creating Chat

The dialog box of the chatbot was developed using some web development tools like HTML, CSS , JavaScript.

F. Pattern Matching

This process included the matching of patterns with the sentences provided by the user and the respective response was returned.

G. *Conversation*

A Chatbot is a program which responds to the queries written in basic english language. It is like communicating to a person in english language. It imitates a normal human.

H. *Architecture*

The architecture of a chabot is depicted in *Fig. 2 .*

V. IMPLEMENTATION

A Chatbot is an application which responds as per query provided by the user. The user interface of the application fetches the input from the user and response from the backend is displayed through the user interface itself. This chatbot was built using some web development tools and

libraries such as bootstrap, javascript libraries to make the UI more interactive and catchy

The elaborate process implementation is stated below:

A. Fundamental Techniques for design

- Creating the bot UI

HTML5 was selected for template creation of the dialog box. For a catchy look some design effects were applied using cascading style sheet, bootstrap .The div tags were used to create the chat box and some CSS attributes were used to give an attractive impression to the user. Some javascript libraries were also imported for background effects.

-Back-end creation

Node js was used for backend programming. It's a bridge between python script and the front end program. Input from the user was taken from the front end, then passed to the Node js and eventually passed to the python script i.e the model.

B. Modules Description

Modules used are stated below as follows :-

- Feature Identification

This was the first step in the entire process. It's used generally to automatically select only relevant features required for the project. It ignored the irrelevant features which resulted in better accuracy for the model. This was one of the enhancements that was done to obtain a precise output. A built-in data analysis tool named Panda was used for data analysis, because of its high performance and easy to use data structures it was a good choice.

a) pd.read_csv()

While using panda the initial step was to get all data from the dataset which was in .csv format, it included irrelevant features too using pd.read_csv() . The entire data set was saved in a data-frame (data structure in panda).

b) CleanedList []

The relevant features extraction was done and saved in an array list .The output from extraction contained a list of objects which had symptoms and diseases as properties. Symptoms ultimately became the features.

- Removal of duplicates

Duplicated removal was also done. This step is often ignored by programmers while developing a bot, but ultimately it impacts the output of the model. Duplicates removal can be of several kinds such as removal of duplicate features, duplicate rows etc.

a) training_data[]

This was the final array in which after all processing the dataset is put into. This training_data[] was used further in the text extraction process and for score calculation as per user input using the Multinomial Naive Bayes model .

- Text Extraction

Text extraction was done in two steps, i.e Stemming and Tokenization.

a) Tokenization

Tokenization was performed to the output after dataset cleaning was done successfully. It is a process of breaking a sentence into compact units. Example: " I am a software engineer " , after applying tokenization the bag of words will be in the form of an array . The array will be like - arr [] = {"I" ,"am" ,"a" ,"software" , "engineer" };

b) Stemming

After tokenization was accomplished stemming was needed to be performed .Since a user can input a sentence with the same meaning in multiple ways. For a single query user can have multiple ways of question formulation. The process to convert a word to its root word is called stemming. This was done using the stemming tool of python. Example - "having " is converted to its root word i.e " have", "sleeping" or "sleeps" to "sleep" etc.

c) NLTK

NLTK is an acronym for natural language tool kit. It was used for categorizing text while working with corpora. Corpora is nothing but a bag of words after tokenization and stemming is accomplished. It consisted of words with their occurrence count. Sometimes words have zero occurrence, which results in zero frequency error. To overcome this problem the proposed system was initialized with 1 as the default frequency. NLTK was used because of its easy-to use and convenient text manipulation using built-in functions like - word_tokenize() for tokenization , stem() under

LancasterStemmer tool for stemming purpose. A bag of words or a dictionary was created named corpus_words [].

- Model building

This chatbot was purely based on the classic algorithm i.e Multinomial naive Bayes Algorithm. It was used in order to classify the instances of problems and label them with classes. But why did we choose this algorithm ? It's model follows probability distribution. Since it works on the principle of independent feature selection, it can perform very well without a huge dataset, i.e it's model works pretty well with a small dataset too.

a) classify()

This function took sentences as a parameter, provided by the user as an input . This sentence was then passed to calculate_class_score() as it's first parameter and the respective scores were returned. This method looped through each and every class from the training dataset and passed the class_name as the second parameter to the calculate_class_score().

b) calculate_class_score()

This function had three parameters, first was the sentence passed by the classify(), other was the class_name which was a particular class in the training data set. This function transformed the sentences into their root word and created a bag of words. Then the stem of the word was checked if it was present in any of the classes i.e class_name. After that the score was calculated for the respective class , the class with highest score was returned.

c)calculate_class_score_commonality()

To get the output in percentage form, this function could have been used instead of calculate_class_score().

C. Figures and Tables

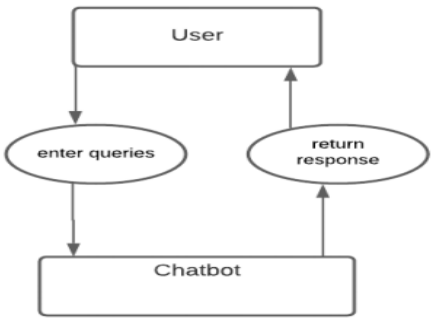

Fig. 1: Use case diagram of the Chatbot

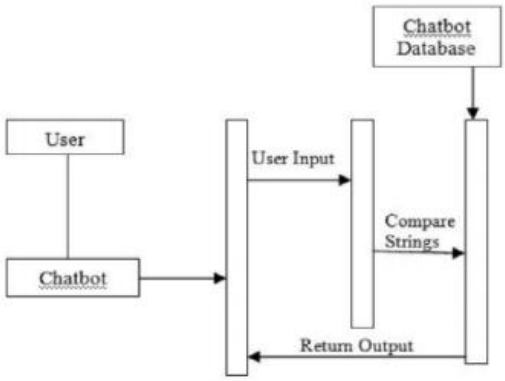

Fig. 2: Sequence Design of the Chatbot

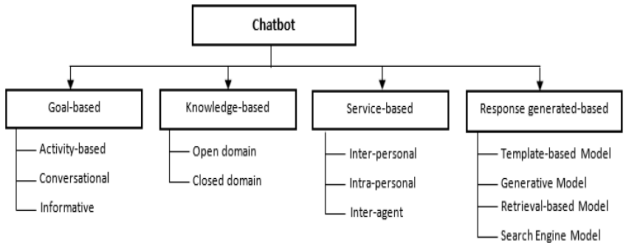

Fig 3: Classification of Chatbots

VI. RESULT DISCUSSION

Easy for users to interact and comprehend. The functioning of the proposed system is quite straightforward which makes it convenient for users as well. Existing chatbots have complicated dataset and entity mapping there is somewhat critical. Removal of the zero frequency error was done in the beginning itself during the text extraction process. Model was fed with an unparsed parameter which reduced the parsing time of the string vector to integer vector, hence the model resulted in response time complexity reduction. Output from the model displayed the class name explicitly, while if the in-built method from the python library was used, output was a matrix which further needed to be evaluated. Time Complexity of the system was reduced to some extent. Existing chatbots use a very huge dataset to train a model which makes it very time consuming and complex. The proposed model was trained using a moderate dataset, because the Multinomial Naive Bayes algorithm can perform well with moderate dataset too. Patient's anxiety will be reduced at home itself. User doesn't need to rush immediately to the doctor, some remedies can be suggested by the bot which will save doctor's as well as patient's time . As many symptoms will be provided by the user the more accurate result will be produced by the bot.

VII. CONCLUSION

A Chatbot has become a very remarkable application of technologies. During this decade chatbots in the market have evolved in many ways. The very initial bots were not much accurate as compared to bots nowadays. Day by day efficiency is increasing. The need of today's time is that chatbots should be user friendly so that anybody can interact with it very easily. Either it is the user interface or the functioning it should be understandable. In this project we went through the design, data cleaning and model training of the bot. This paper proposes one technique to reduce time complexity to some extent. More research can be done in this field by the researchers in order to make it more efficient. Many chatbots have been developed so far but still improvements can be done.

REFERENCES

[1] Mohammad Nuruzzaman ,Omar Khadeer Hussain (2018) A Survey on Chatbot Implementation in Customer Service Industry through Deep Neural Networks. 2018 IEEE 15th International Conference on e-Business Engineering (ICEBE).

[2] Tarun Lalwani, Shashank Bhalotia (2018) College Information Retrieval System. (IJIRCST) ISSN: 2347-5552, Volume-6, Issue-3, May 2018

[3] Balbir Singh Bani, Ajay Pratap Singh (2017) Chatbot for college QnA using A.L.I.C.E. IJNTR Jan 2017

[4] Bing Liu,gokhan Tur,Larry Heck (2017) Dialogue Learning with Human teaching and feedback. NAACl HLT, 2018.

[5] M.Dahiya (2017)A tool of conversation. IJSE 2017.

[6] Io, H. N., & Lee, C. B. (2017). Chatbots and conversational agents: A bibliometric analysis. 2017 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM).

[7]. Li, C.: "Algorithm of Web Session Clustering Based on Increase of Similarities", Proceedings of International Conference on Information Management, Innovation Management and Industrial Engineering, pp. 316–319. IEEE, Los Alamitos (2008)

[8].G. Poornalatha and P. S. Raghavendra, " Web User Session Clustering Using Modified K-means Algorithm," First International Conference on Advances in Computing and Communications (ACC – 2011), CCIS(191),Springer-Verlag, pp.243-252, 2011

[9]. Xu, J.-H., Liu, H., "Web User Clustering Analysis based on KMeans Algorithm", International conference on Information, Networking and Automation,(ICINA), pp. V26–V29. IEEE, Los Alamitos 2010

[10]. W. Wang, O. R. Za¨ıane, "Clustering web sessions by sequence alignment "University of Alberta Edmonton, Alberta, Canada.

[11] Hazarath Munaga, J. V. R. Murthy, N. B. Venkateswarlu, "A Hybrid Trajectory Clustering for Predicting User Navigation", International Journal of Recent Trends in Engineering"

[12]. Chung, Kyungyong, and Roy C. Park. "Chatbot-based healthcare service with a knowledge base for cloud computing." Cluster Computing22, no. 1 (2019): 1925-1937.

[13]. Divya, S., V. Indumathi, S. Ishwarya, M. Priyasankari, and S. Kalpana Devi. "A self-diagnosis medical chatbot using artificial intelligence." Journal of Web Development and Web Designing 3, no. 1 (2018): 1-7.

[14]. Hoermann, Simon, Kathryn L. McCabe, David N. Milne, and Rafael A. Calvo. "Application of synchronous text-based dialogue systems in mental health interventions: systematic review." Journal of medical Internet research 19, no. 8 (2017): e267.

[15]. Kavitha, B. R., and Chethana R. Murthy. "Chatbot for healthcare system using Artificial Intelligence." (2019).

[16]. Rarhi, K., Bhattacharya, A., Mishra, A., & Mandal, K. (2017). Automated Medical Chatbot.