# Multi Domain CHAT BOT Using Random Forest classifier and Machine Learning

Jyoti Dixit, Suman Kumar Swarnkar, Lalit P Bhaiya
(Department of Computer Science & Eng, Bharti College of Eng. & Tech. Chhattisgarh, India.)

***Abstract :*** Chat bots are intelligent systems that perceive users tongue queries and respond consequently during a speech, that is that the focus of this study. it's a lot of sort of a virtual assistant, folks want they're talking with real person. They speak an equivalent language we tend to do, will answer queries. In multidomain, at client care centers and enquiry desks, human is meager and typically takes while to method the one request which ends in wastage of your time and conjointly cut back quality of client service. the first goal of this chat larva is, client will act with mentioning their queries in plain English and also the chat larva will resolve their queries with acceptable response reciprocally. The planned system would facilitate replicate the client service expertise with one distinction that the client would be interacting with a larva rather than a true person and nevertheless get the queries attended and resolved. It will extend lifestyle, by providing solutions to assist desks, phone respondent systems, client care centers. This paper explains the dataset that we've ready from FAQs of multidomain websites, design and methodology used for developing such chat larva. Also, this paper discusses the comparison of seven classification algorithms used for obtaining the category of input to speak larva.

***IndexTerms* - Chat bot, multidomain, classification, NLP, vectorization**

## INTRODUCTION

Multidomain play a crucial role in each country's economic development. In daily life, everyone wants multidomain. however most of the folks, particularly the first-timers, struggle to grasp numerous procedures and processes needed to urge their work done at the multidomain and avail of its totally different product and services. presently multidomain have their own web-sites, mobile applications and facilities like web multidomain, mobile multidomain however generally, these sources are often a small amount overwhelming for many of the users World Health Organization are either not well versed with technology or in some cases wherever the data is simply too scattered to go looking for simply. There are differing types of platforms provided by totally different multidomain however folks face issues accessing them (different GUIs, an excessive amount of navigation). though client Care centers are obtainable, there are heap of wait times and redirection in some cases, deed the client with no selection however to expertise right smart delays obtaining a straightforward informational question resolved. folks have queries regarding numerous multidomain policies, loans, fastened deposits. This leads to excess crowd in multidomain for inquiry. Multidomain conjointly face issues resolution perennial queries of consumers. this can be time intense and multidomain workers gets annoyed. men and cash get wasted for separate inquiry counter.

### A. **Basics of Chat Bot**

A chat larva may be a informal agent that interacts with users during a sure domain on sure topic with tongue Sentences. unremarkably a conversation larva works by a user asking a matter or initiating a replacement topic. Chat bots are often known as software package agents that simulate associate degree entity typically an individual. These are the software package with AI that permits them to know users input and supply meaty response victimization predefined mental object.

### B. Chat Bot for Multidomain

Developing a conversation larva can offer a wise resolution to unravel these queries, offer info as and once needed, improve service and increase variety of consumers. It removes human factors enclosed in organization and may provide 24/7 hours service to extend productivity. we tend to will offer a conversation larva interface for patrons that may well be obtainable on the online and on any hand-held devices. Customers will mention their queries in tongue and also the chat larva will answer them with correct answer. planned chat larva application is well accessible to client thereby resolution redundant queries anyplace anytime. As there'll be quick response for inquiry, this can be time saving for each multidomain and customers. The planned system would be a stepping stone in having in situ associate degree intelligent question handling program that might in next phases not simply respond however self-learn to enhance itself thereby increasing not simply the standard of client service however conjointly reducing human load, increase in productivity and after all increasing variety of glad customers.

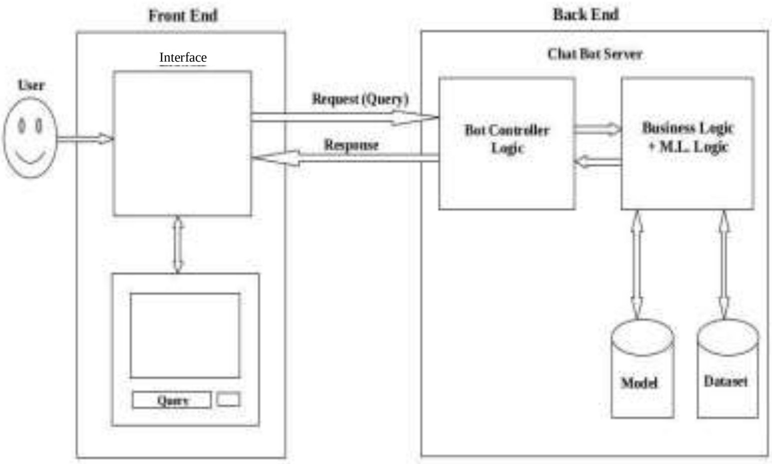
## RELATED WORK

Current chat bots are developed victimization kind of ways like rule based mostly wherever rules are hard-coded in code, AI based mostly bots, pattern-based which may handle solely mentioned patterns for retrieving answer. There are frameworks obtainable for developing chat bots however they conjointly use either rule-based or pattern-based techniques. In rule-based chat bots that are best to create, one has to be compelled to write rules like If X then Y else if A then B etc. So, if there are a hundred situations, developer must write a hundred rules for every of the situations. The volume, selection and complexness of knowledge makes such techniques inefficient. Its nearly not possible to write down rules and/or patterns for massively obtainable information. AI based mostly bots are designed on IP and milliliter. they're supported human capability of learning info however with a lot of potency. tongue process (NLP) are often used wherever predefined or static rules, patterns might not work.

## ARCHITECTURE

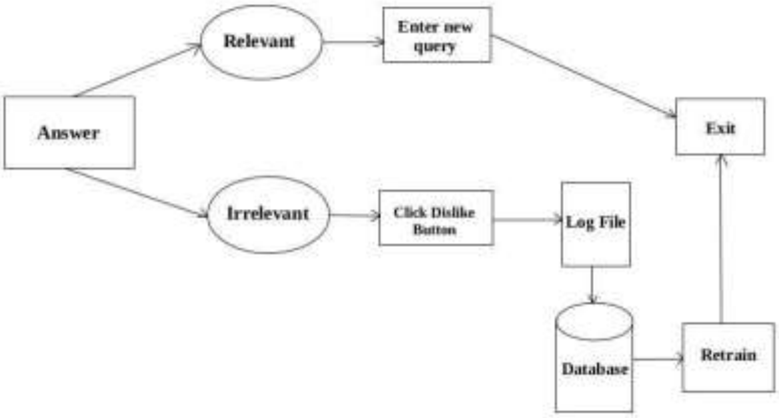
### A. Multidomain Chat Bot

User can act with the system victimization net application. He can enter his question in text box provided on side of this net application. Once he presses Enter button or submit the question, this request is going to be handled by larva controller logic.



The larva controller logic contains implementation of Flask framework for handling user requests and causing answer to those queries as response. Then, the question is going to be sent to Business logic and Machine learning logic. Business logic contains pre-processing of user input question victimization tongue process (NLTK library) and its vectorization. IP can tokenize the question, take away excess areas, stop-words and so extract lemmas for every token. Then this text-format question is going to be reborn to vectorized format victimization vectorization. Now, victimization milliliter logic, classification algorithmic program is going to be applied to the present remodeled question to search out the category it belongs to. Classification algorithmic program are going to be applied supported the previous saved model dead on train information. All queries from input file having category up to retrieved category are going to be fetched and trigonometric function similarity are going to be applied to those. in step with similarity values we tend to get, most similar answer is going to be came to user as response.

### B. Feedback System



There are things wherever chat larva cannot provide right answer or cannot have answer to asked question as a result of the question is out of info. For such circumstances, we've developed feedback mechanism for our chat larva. net application can have

Dislike button alongside submit button. If just in case user isn't glad with the solution provided by system, he will press this Dislike button. Then log file are going to be generated for this question or the query are going to be inserted into already created log file. Developer comes into image currently for handling such things. He can check categories for these queries, enter right answers and can retain the classification model. in order that whenever user enters an equivalent question next time he can get correct answer. during this approach chat larva can improve its accuracy and dataset.

## IMPLEMENTATION

1) Preparing Data Set: we've ready our information set as queries and answers that individuals typically raise to multidomain workers, at client care centers or inquiry desks. we've referred totally different multidomain websites and picked up FAQs as our information. we've used totally different net scrapping tools for this task. Distribution of queries in data-sets;

2) Pre-processing: we've used NLTK library for tongue process. As user input are going to be in English statement, to let machine perceive this language we tend to use tongue process. To decrease any process and removing ambiguity caused because of use of same word in several forms, we've done this pre-processing. Steps enclosed during this task are:

Removing punctuation marks and extra spaces

- Tokenization - we've used tokenization to come up with sequence of words from users input question. Removing stop words - Most of the common words like want, are, can, that we tend to dont have to be compelled to be thought-about whereas process is removed for rising the performance of system.

- Lemmatization - we've used WordNet Lemmatizer for obtaining lemma (root style of the word) of every token. e.g. processing and process ought to be thought-about equal whereas process. So, for obtaining process from processing, lemmatization is employed.

3) Vectorization: we've reborn our text information to vectorized format victimization Bag of Words (BOG) construct. bathroom may be a methodology for making ready text for input to our machine learning algorithmic program. bathroom model develops a vocabulary from all of the documents and so model every document by enumeration variety of times every word showing in various document.

4) Classification: because the information set will increase, it takes longer to search out similarity between users question and also the queries from massive information set and come back the solution. therefore we've used classification to enhance the potency by reducing the latent period needed to urge the solution. we've used Scikit-learn library for implementing these classifiers. Scikit-learn is tool for data processing and machine learning in Python. As an area of literature survey and initial coaching we've chosen following set of classifiers to settle on the simplest acting one because the final classifier for the chat larva.

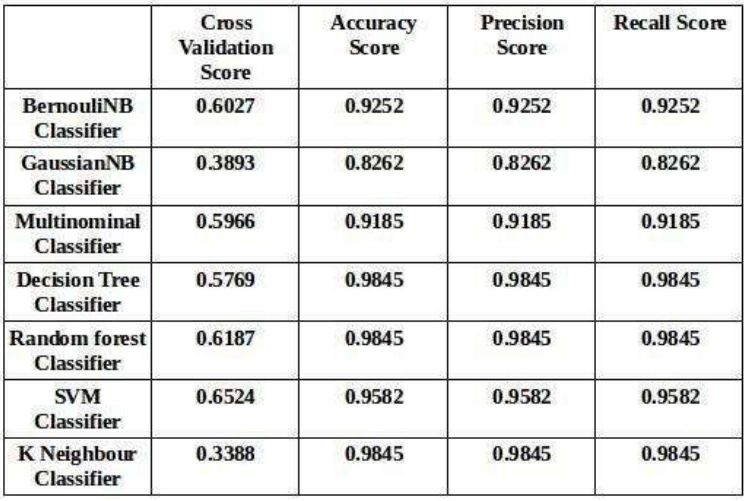
- Decision Tree classifier
- Bernoulli Naive Bayes Classifier
- Gaussian Naive Bayes Classifier
- K-nearest neighbor classifier
- Multinomial Naive Bayes classifier
- Random Forest classifier
- Support vector machine

Also, for optimizing the algorithm’s performance according to our data set, we have implemented parameter optimization. There are two approaches for implementing parameter optimization -

- Grid search - Grid search is simply exhaustive searching approach. In grid search, it is needed to manually specify subset of hyper-parameter space of a learning model. Hyper-parameters are the parameters that are not directly learned within estimators.
- Randomized Search - As grid search is exhaustive and therefore expensive. So, in randomized search, it samples parameter settings a fixed number of times that are more effective. We have used Randomized Search approach in our case.

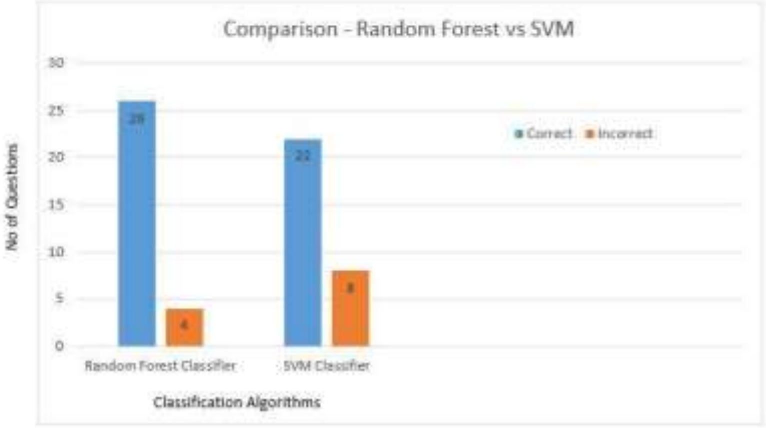
5) Developing learning model: In this phase, we have combined Natural Language processing, Vectorization and classification algorithms all together and save this model for further use. So, whenever the new query comes to system, we will just fetch this saved model, test this query on that model and get its class. In this way, we don’t need to train model every time for each new query, thereby reducing the processing time.

6) Testing model: Checking for cross-validation score and precision and recall score of each classification algorithm, so that we can choose best for final use. Following is the table containing scores of each algorithm:



| | Cross Validation Score | Accuracy Score | Precision Score | Recall Score |
|---|---|---|---|---|
| BernouliNB Classifier | 0.6027 | 0.9252 | 0.9252 | 0.9252 |
| GaussianNB Classifier | 0.3893 | 0.8262 | 0.8262 | 0.8262 |
| Multinominal Classifier | 0.5966 | 0.9185 | 0.9185 | 0.9185 |
| Decision Tree Classifier | 0.5769 | 0.9845 | 0.9845 | 0.9845 |
| Random forest Classifier | 0.6187 | 0.9845 | 0.9845 | 0.9845 |
| SVM Classifier | 0.6524 | 0.9582 | 0.9582 | 0.9582 |
| K Neighbour Classifier | 0.3388 | 0.9845 | 0.9845 | 0.9845 |

7) Choosing best approach: According to scores of above tables, 2 most accurate algorithms are - Random Forest classifier and Support Vector Machine classifier.

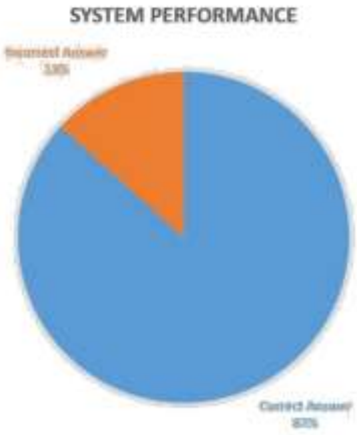


8) Query mapping and getting answer (Using Cosine similarity): Once classifier gives us the class, we extract all questions that have this class from our data set. We check for cosine similarity of user's query with these extracted questions. Then answer of the most similar question is chosen as response to user's query and is returned to him. This bot is closed domain i.e. restricted to multidomain only. We have set a threshold on values of cosine similarity measure for handling queries that are out of domain.

## EXPERIMENTAL RESULTS

**Experiment 1**

We have tried different queries to validate the implementation of Multidomain Chat Bot. In this experiment, we have entered queries which are similar to the questions present in our data set. The analysis of the result is shown below:



## CONCLUSION

The planned system would be a stepping stone in having in situ associate degree intelligent question handling program that might in next phases not simply respond however self-learn to enhance itself thereby increasing not simply the standard of client service however conjointly reducing human load, increase in productivity and after all increasing variety of glad customers.

## FUTURE SCOPE

- Widening the domain
- Intelligent answers constructed by combining not just the existing list of FAQs but also from various other sources like internet, databases and other sources of data
- Providing close suggestions
- Intelligent representation of response images, links
- Combining semantic similarity along with cosine similarity
- Showing account related information using Multidomain’s Gate-way

## REFERENCES

[1] Automatic Extraction of Chatbot Training Data from Natural Dialogue Corpora, Bayan AbuShawar, Eric Atwell

[2] Chatbot Evaluation and Database Expansion via Crowdsourcing, Zhou Yu, Ziyu Xu, Alan W Black, Alexander I. Rudnicky AI BASED CHATBOT, Prof.Nikita Hatwar, Ashwini Patil, Diksha Gondane Data Mining: Concepts and Techniques Jiawei Han and Micheline Kamber

[3] H. Kopka and P. W. Daly, Data Mining Practical Machine Learning Tools and Techniques Ian H. Witten Eibe Frank Mark A. Hall

[4] https://www.cse.iitb.ac.in/ bibek/WriteUP2016:pdfhttp : ==cs224d:stanford:edu=

[5] https://scikit-learn.org

[6] http://www.nltk.org

[7] http://www.wikipedia.org

[8] https://chatbotsmagazine.com/the-complete-beginner-s-guide-tochatbots-8280b7b906ca.i2zgql2op

[9] https://www.quora.com/What-is-the-best-way-to-

[10] learn-and-write-a-AI-Chat-bot

[11] https://chatbotslife.com/ultimate-guide-to-leveraging-nlp-machine-learning-for-you-chatbot-531ff2dd870c.5mcveo57b

[12] https://apps.worldwritable.com/tutorials/chatbot/

[13] http://machinelearningmastery.com/

[14] http://flask.pocoo.org/

[15] pandas.pydata.org

[16] https://www.pandorabots.com/

[17] http://www.wildml.com/2016/04/deep-learning-for-chatbots-part-1-introduction/

[18] https://stanfy.com/blog/advanced-natural-language-processing-tools-for-bot-makers/