Conversational AI - A Retrieval Based Chatbot

Aiswarya Surendran, Reena Murali and Remesh K. R. Babu

August 10, 2020

Conversational AI - A Retrieval Based Chatbot

*

Aiswarya K. Surendran
Computer Science and Engineering
GEC Palakkad
Palakkad, India
aiswaryasurendrank@gmail.com

Reena Murali
Computer Science and Engineering
GEC Palakkad
Palakkad, India
reenamurali@gecskp.ac.in

Remesh Babu K. R
Information Technology
GEC Palakkad
Palakkad, India
remeshbabu@gecskp.ac.in

***Abstract*—Conversational AI can be simply defined as human-computer interaction through natural conversations. This may be through a chatbot on a website or any social messaging app, a voice assistant or any other interactive messaging-enabled interfaces. This system will allow people to ask queries, get opinions or recommendations, execute needed transactions, find support or otherwise achieve a goal through conversations. Chatbots are basically online human-computer dialog system with natural language. Currently, advancements in natural language processing and machine learning mechanism have improved chatbot technology. More commercial and social media platforms are now employing this technology in their services. Organisations demands artificial intelligence based improvements in chatbot adoption and thus it became one of the hot research. In this work, a task-oriented retrieval based chatbot has been proposed on a bus ticket booking domain which is built using a Deep Neural Network. The sequence of questions that are being asked by multiple users with different kinds of personas are taken as the input for the system. And accordingly the retrieval based system produces meaningful responses. The generated responses are evaluated manually. The results show that the generated answers are meaningful in most of the cases.**

***Index Terms*—Chatbots, Retrieval-based model , neural network, Deep Learning**

I. INTRODUCTION

A chatbot is a computer program that allows humans to interact with technology using a variety of input methods such as voice, text, gesture and touch. For several years chatbots were typically used in customer service environments but are now being used in a variety of other roles within enterprises to improve the customer experience and business efficiencies. Chatbots are known by a variety of different names such as a conversational AI bot, AI assistant, intelligent virtual assistant, virtual customer assistant, digital assistant, conversational agent, virtual agent, conversational interface and more. But just as chatbots have a variety of different names, they also have varying degrees of intelligence. A basic chatbot might be little more than a front-end solution for answering standard FAQs.

Chatbots built using some of the bot frameworks currently available may offer slightly more advanced features like slot filling or other simple transactional capability, such as taking pizza orders. But, only advanced conversational AI chatbots have the intelligence and capability to deliver this sophisticated chatbot experience. At the same time, It is something that most of the enterprises are looking to deploy.

A Conversational AI can be implemented in different ways. Each of that approaches have their own merits and demerits. Let us discuss about some of the approaches that we generally use for building conversational systems.

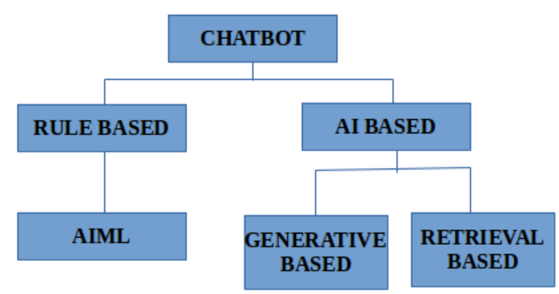

Fig. 1. Overview of chatbot

A. RULE-BASED SYSTEM

Rule based system[1] works on the basis of a set of defined rules. The system usually trained in such a way that, if a certain input had given by the user, the response should be a particular one. Actually here a transformation is taking place from input to output based on defined set of rules. However this rule based system are not able to produce output when a query which are not defined in the rule set is asked.

B. AI BASED SYSTEM

Artificial Intelligence Based System generally uses the concept of machine learning approaches to produce responses for the input which is given by the user. As this system mainly depends on ML approaches, It is termed as more efficient than that of Rule Based Systems. Basically AI based system can be subdivided in to two types. Namely Retrieval based model and Generative based model.

Retrieval based system maintain a database of predefined responses. It is some how similar to the rule based system, but more efficient than that of rule based. Here when user ask a query, the system actually selects the best response by means of simple algorithms like keywords matching or it may require also more complex processing with machine learning or deep learning. Basically this system need a-lot of data pre-processing. The databases should be manually updated at a particular interval to avoid the problem of outdation. The retrieval-based bots usually come up with a set of written responses to minimize the grammatical errors and to improve coherence and avoid a situation a system can be hacked to respond in less appropriate ways. Basically this type of chatbots are very suitable when we are dealing with domain specific conversational systems. On the other hand Generative based system is trained in such a way that it can also produce response if a new query arrived. While the other two methods produces output on the basis of the rule defined or by considering matching from database, generative model needs large amount of data to get trained and from that the system try to generate new responses for a new query. When we are using generative model, there is high chance of producing grammatically error sentences as it have high freedom in generating responses. Generative system mainly use the concept of supervised learning, reinforcement learning, and adversarial learning for model building.

II. RETRIEVAL BASED CONVERSATIONAL AI

The proposed system is a retrieval based system which particularly concentrate on a specific domain. A closed domain of Bus ticket booking query is taken for implementation.Retrieval based bots works by using the principle of directed flows or graphs. Basically such type of bots are trained to rank the best response from a finite set of predefined responses. The responses here are entered manually by the developer, or based on a knowledge base of pre-existing information. The system is trained in such a way that there will be a set of questions that usually ask by the user multiple times in multiple ways and also set of corresponding responses. Retrieval based bots are the most common types of chatbots that we see today. It allows bot developers to control the experience and match it to the expectations of the customers. It work best for goal-oriented bots in customer support, lead generation and feedback. In retrieval system we can even decide the tone of the bot, and design the experience, keeping in mind the customer's brand and reputation. Figure 2 shows the basic architecture of the retrieval based chatbot system.Given a user input utterance as the query, the system basically searches for candidate responses using any of the matching metrics by considering the knowledge base. The core of retrieval-based conversational systems is formulated as an identical problem between the query utterance and the candidate responses. The model will find out the context of the query by identifying it's main intent. A matching is done between the query fed by the user and the set of queries that are already stored in the repository. A typical way for matching is to measure the inner-product of two representing feature vectors for the queries and the candidate responses. A series of matching methods are often applied to short-text conversations for a retrieval-based system. Basically, these methods model sentences using convolutional(CN) or recurrent networks(RN) to construct abstractive representations. A series of matching metrics have been proposed for retrieval using deep neural networks. Usually, sentences are compared in a pairwise matching style via word-by-word matchings, known as sentence pair modeling. The chain based matching is also demonstrated to be useful by mixing sentence information as a chain sequence. In chain-based matching, modeling the second sentence is not blind to the modeling of the first sentence. Although not all of those methods are originally designed for conversation, they are effective for short-text matching tasks and are included as a strong baseline for retrieval-based conversational studies.

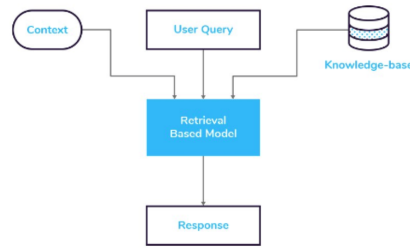

Fig. 2. Retrieval Based Chatbot

III. RELATED WORKS

The conversational agent[2] proposed by Oriol Vinyals *et al.* used sequence to sequence framework. Their model converses by predicting the next sentence given the previous sentence or sentences in a conversation. The strength of their model is that it can be trained end-to-end and thus requires much fewer hand-crafted rules. Wu *et al.*[3] Response selection with topic clues for retrieval-based chatbots. They used Long-Short Term Memory(LSTM) neural network model to improve the performance of QA systems. Language models such as GloVe are also considered. Using Survival Motor Nueron(SMN) network, their result shows over 6% and significantly outperforms the state-of-the-art methods. The approach[4] proposed by Braden Hancock *et al.* deals with a self-feeding chatbot, a dialogue agent with the ability to extract new training examples from the conversations that it participates in. Here the user's responses become new training examples to imitate on the next step. When the agent believes it has made a mistake or error, it asks for a feedback; Basically learning to predict the feedback that will be given by the user improves the chatbot's dialogue abilities also. In Lison's work[5]it is shown a mechanism for representing the underlying structure of probabilistic models for dialogue generation using probabilistic rules. This approach has been implemented in an oral dialogue system for human-robot interaction and has been validated in a policy-learning task based on a Wizard-of-Oz dataset. The System model of Dusek *et al.* [6] present a new system based on natural language for oral dialogue generation capable of incorporate in its behavior the manner in which users use to talk or communicate, providing appropriate contextual responses. The generator system is based on recurrent neural networks and the sequence-to-sequence approach. The work of Jason D. Williams*et al.*[7] presents a model for end-to- end learning of task-oriented dialog systems. The main component of the model is a recurrent neural network (an LSTM), which maps from raw dialog history directly to a distribution over system actions. The LSTM automatically infers a representation of dialog history, which relieves the system developer of much of the manual feature engineering of dialog state.

IV. CONVERSATIONA AI - RETRIEVAL BASED MODEL

The basic aim of building a Deep Learning[8] model using the input questions to drive the conversation in a more meaningful way is considered while building the system model. Here the proposed system works on a manually developed Bus ticket Booking querying dataset and it produces relevant responses based on the pre-defined repository. Basically Keras is a minimalist Python library that we use for deep learning. Here a Deep Learning model from keras called Sequential is used for implementing the model. Fig 3 shows the basic architecture of the proposed retrieval based chatbot system. The overall system architecture can be classified in to various sub modules as shown. Data is considered as the basic key

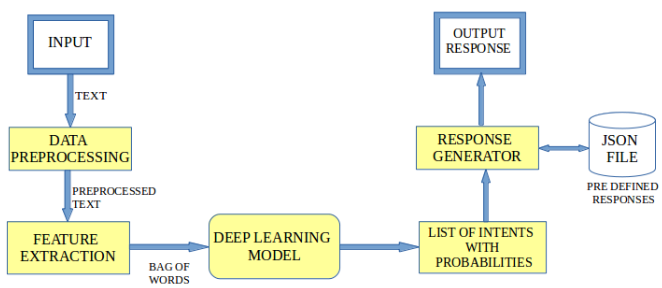

Fig. 3. System Overview

to a chatbot system. If we want our chatbot to be truly conversational, we should consider it's dataset as a crucial element. To build a conversational system with moderate intelligence is very challenging as it requires abundant dialogue data and interdisciplinary techniques. Here a data set is created manually in order to get more accuracy. As the system is a retrieval based one, it works mostly on a closed domain. As mentioned earlier a closed domain of ticket booking queries are considered. Data set preparation was the first task that had done as a part of the system implementation. We can even get data set online, but the problem is that, most of the datas are not in a format that we are looking for. So in such scenario, it is better to create data set of our own. It seems to a tiring task at beginning, but at the end it will really help in generating relevant responses. At a particular interval of time, the data set is to be updated again and again and should retrain the model to improve accuracy. Otherwise the data get outdated. When user keeps on asking a query which was not defined in the repository multiple times, it is better to add the responses of such queries to the data set manually. This process will definitely improves the efficiency of our chatbot system thereby reducing the number of unanswered questions. Dataset which is used here is a bunch of JavaScript objects that lists different tags that correspond to different types of word patterns and is stored as a Json file which contains the intents. A sample of possible questions and answers of the dataset are shown here.

Query: *"How you could help me?"*
Response: *"I can help you to book your Bus tickets"*

Query: *"Does booking online cost me more?"*
Response: *"Not at all! The price of the bus ticket is the same as you would get from the bus operator too."*

After dataset creation, various preprocessing tasks are carried out for cleaning the data. Preprocessing task is termed as an important task while we are dealing with any NLP tasks. Here the data that we are considering are pre-processed to remove any content that does not have any useful information in them. Task of preprocessing is done here in the complete dataset which contains various user input queries and responses and also the same preprocessing task are carried out when a particular input is given by the user. Table I shows the tools and libraries used in the text preprocessing phase. Basically we used pandas library functions for reading the dataset. Then the sub tasks in the text preprocessing phase has been carried out by Natural Language Tool Kit (NLTK), and some string handling functions in Python.

TABLE I
TOOLS AND LIBRARIES USED IN TEXT PREPROCESSING TASK

Task	Method Used
Tokenization	NLTK
Lemmatization	NLTK
Conversion to Lowercase	Python String Handling Functions

Feature generation is the next task that we need to consider. After cleaning the data, It should be converted in to such a way a machine can understand. For that purpose we use this feature generation techniques. We can perform feature generation in different ways. Here the concept of Bag Of Words(BOW) are considered for generating features from the input text. In this model, a text such as a sentence or even a document is represented as the bag (multiset) of its corresponding words, with out considering the grammar and even word order but keeping multiplicity. The output from this feature generator will be given as the input to the deep learning Model to predict the response.

A DL based retrieval chatbot system is implemented here. In problem definition, we already mentioned that our primary aim is to build a Deep Learning system that takes input as a set of queries that are being asked by the user with multiple personas and produces the most accurate responses and to drive the conversation in a more meaningful way. A deep learning model from keras called Sequential is used here for building the model. From the figure 4, it is understood that

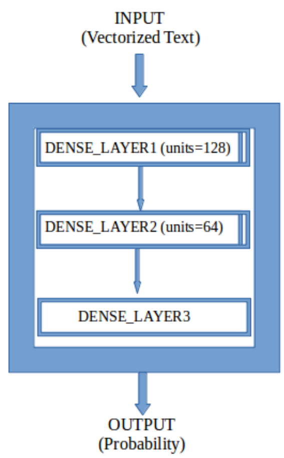

Fig. 4. Sequential Model of Keras

there are 3 layers for the DL model. Also 2 dropout layers are also there in between every Dense Layer. In this 3 Layer concept, the first one have 128 neurons, the second one have 64 neurons, and the third one have the number of intents as the number of neurons. Basically the aim of this network is to predict which intent is to be choosed if we are given some input data. The input to the DL Sequential model will be vectorized text. On the previous module, we already discussed about feature generation. Here the embedding purpose are carried out by Bag Of Words(BOW) concept. After converting the text in to the bag of words,we get the corresponding index for the each words used in the training data. This will pass to the DL model. The data will pass through each of the dense layer. Then the model is trained with Stochastic gradient descent(SGD). Stochastic gradient descent is more efficient than normal gradient descent. The number of layers and the neuron numbers are considered after performing hyper parameter tuning. The system is even trained with and without dropout layers also. After training the model , the whole thing is turned into a numpy array and saved. We can use this model for the prediction of the most accurate intent. For the purpose of prediction, an error threshold of 0.25 is used. This is use just to avoid too much overfitting. So when the input is passed through the DL model, it usually generates the list of intents from the repository and it's corresponding probabilities along with their likelihood of matching the correct intent. We then filter the list again by using the concept of error thresh hold value. Only the intents that have a probability more than that of the threshold should be considered. For model building, both ReLu and softmax activation functions are used. The ReLu function is used both in input layer and the hidden layer. The softmax function is used in the output layer also. We used ReLu in input and hidden layer because it can usually lead to less dense solutions. Many other activation functions, such as the sigmoid, will always output a non-zero value. This means every neuron will be considered at each pass, even if we do not need it. The ReLU function does not always output a non-zero, so it results in less neurons being utilized and less dependence between features. The softmax function is used in output layer because the purpose of this function is to give a probabilities of certain classes occurring. Here also our aim is to get the probability of each of the intents in the dataset by comparing it with the inputted query. Table II shows the implementation details of the deep learning model.

TABLE II
THE DEEP LEARNING SEQUENTIAL MODEL

Parameter	Value
Layers	Dense
Activation	ReLu and Softmax
Optimizer	SGD
Threshold	0.25
Epochs	200
Batch Size	5

The response generation is the final task of the system model. The DL model gives us list of intents after filtering it with error threshold along with their corresponding probabilities. It is the duty of response generation module to select the one from that list that is very much suited for the inputted query. This module takes the output from the DL module as it's input. Then these list of intents are checked with the repository(here Json file of intents) and the one with highest probability that matches perfectly well for the input is produced as the final response.

A. *Special Features*

Special features module is added to the proposed system in order to make the chatbot more user friendly. We know that the users will become more attracted if the system will have added features in it. So some features like that are also added. The features include Spell checker and Speech-to-text converter. The features are added in the input giving part. The block diagram showing the concept of these features is shown on Fig 5.

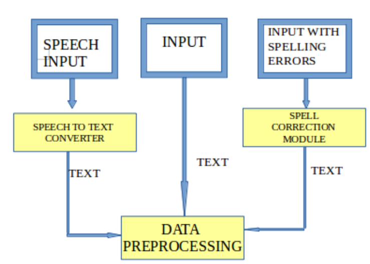

Fig. 5. Special Feature Module overview

We know that chat language is far from the standard English. Different user follow different manners while chatting. Even there is high chance of using shortforms and also there is high chance of typing something with spell errors also. One of the real challenges to any sort of chatbot lies in understanding the spirit of the input that had given by the user, even if its not spelled correctly. If the system is getting a wrong input, there is high chance of creating mistakes while generating responses also. So in order to solve that, a spell checking module is implemented. When the user give an in-correcly spelled input to the system, the system will give the user a set of corrected input. From that set, user can select the most appropriate one for his need. That selected option is then considered as the input and it is then fed to the data pre-processing step. Then the whole process works as usual. Here spell checking module is implemented by using NLTK spell checking module and a dictionary concept which contains a large collection of correctly spelled words.

Some user loves to chat with the system by giving speech inputs. For such users, we added this concept. Here by using a microphone, speech inputs can be given to the system. System will identify the source of this speech, and hence converts this speech to text. Basically different Speech Recognition API's are available now and Google Speech Recognition API is used here to convert the input audio to text.

V. EVALUATION FOR CONVERSATIONAL AI

Basically Automatic evaluation is very crucial for language generation tasks, while most existing metrics evaluate generated sentences by measuring word overlap, referring to ground truth sen- tence(s). For retrieval-based conversational systems, traditional in- formation retrieval evaluation metrics such as precision@n, mean average precision are applicable. For generation- based conversational systems, since there is no specific eval- uation measurement for dialogues, metrics for machine trans- lation (BLEU) and/or summarization (ROUGE) are "bor- rowed" in the majority of conversational studies to evaluate the quality of responses.

VI. RESULT AND ANALYSIS

Various experiments are done in the proposed system model in order to get a system with good accuracy value. As we are dealing with a retrieval based chatbot, the test accuracy can be manually calculated by randomly giving some inputs to the system and analyzing the responses. For most of the input that we are giving, it generated good responses. If an input which is not there in the repository is asked, it will give responses like "Sorry, can't understand you", "Please give me more info", "Not sure I understand". So in-order to improve it's efficiency,as mentioned earlier we should keep on updating the repository with new inputs and it's responses. The dataset hence keep on growing. Re-train the system again and then check again.

Basically Automatic evaluation is very crucial for language generation tasks, while most existing metrics evaluate generated sentences by measuring word overlap, referring to ground truth sentence(s). For retrieval-based conversational systems, traditional information retrieval evaluation metrics such as precision@n, mean average precision are applicable. For generation- based conversational systems, since there is no specific evaluation measurement for dialogues, metrics for machine translation (BLEU) and/or summarization (ROUGE) are "borrowed" in the majority of conversational studies to evaluate the quality of responses. Here the accuracy of the system are manually calculated by using accuracy @ n concept. It is done by asking the queries randomly in different manner at various level n. For most of the input that had given, it generated good responses when n is small value. The average value of accuracy at different levels are taken as the system accuracy of the proposed system. In such a way the proposed system shows 83.6% accuracy. Here the value of n is taken randomly. We can even ask more questions by changing the value of n. But as it is a retrieval based model, when the value of n is increased ,there may be a loss in accuracy. As testing accuracy purely depends on the pre-defined repository, It is also better to consider the training accuracy. Here we used complete dataset for training. Then only, It produces responses for all those mentioned datas on the data-set. The details regarding the accuracy at different levels are clearly mentioned on the Table III.

TABLE III
SYSTEM ACCURACY

Value of n	Accuracy(%)
10	80
15	86
20	85

A. Results

Different experiments are carried out on the model to analyze the change in accuracy by changing the parameters. The details of that experiments and it's corresponding results are also mentioned in the coming sections.

1) Epoch Analysis: The influence of epochs in training accuracy rate and training loss rate are considered here. By changing the number of epochs, the system is trained and noted the difference in loss rate and accuracy rate. It is based on the findings that had get through these experiment, the number of epochs were fixed. The results that had obtained through the experiments are represented as graph.

Training loss can be defined as the error on the training set of data. From the Fig 6, we can see the difference in loss rate with the change in number of epochs taken. When we take number of epochs as 20, The loss rate is keep on changing at each epoch and when number of epochs is increased, the loss rate keeps fluctuating and when a particular point of about 200 is reached, then the loss rate keeps on taking the same value. The accuracy of the model are also noted. Like the same way Training accuracy with the difference in number of epochs was also considered. Training accuracy can be usually as the accuracy you get if you apply the model on the training data. The training accuracy also changes with change

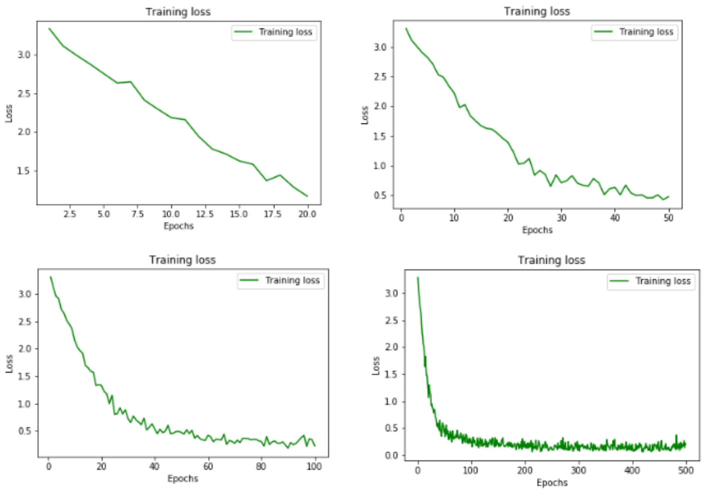

Fig. 6. Training Loss v/s Epochs

in number of epochs. From the Fig 7, We can see that the

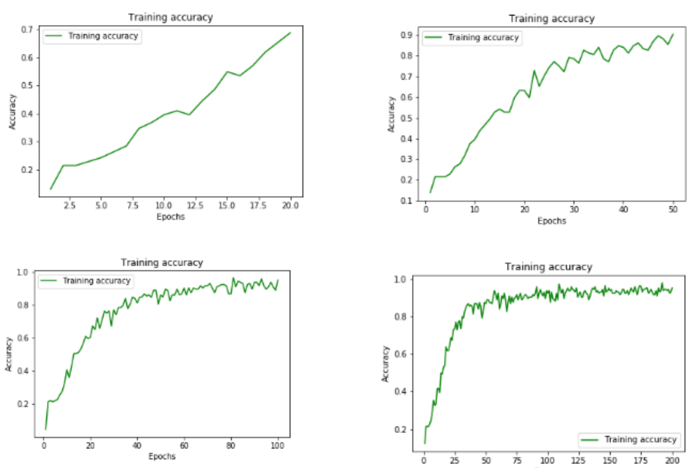

Fig. 7. Training Accuracy v/s Epochs

accuracy of the system keep on increasing with the increase in number of epochs. After a particular value of epoch(here after 100) the accuracy is high and it tries to remain in that point itself without much falls. The accuracy of the model with different number of epochs are clearly mentioned the table By considering the results of experiments and tl

TABLE IV
ACCURACY

No of Epochs	Accuracy(%)
20	68
50	90
100	93
200	95
500	90

accuracy noted, it is evident that it will be better to set epoch to a value greater than 100 and lesser than 500 to get good results. So by considering that the epoch number is set to a particular value. Basically here we done a hyper parameter tuning. Hyper parameter is very important term that we need to consider while dealing with network building. It is termed as the variables which determines the network structure(Eg: Number of Hidden Units) and the variables which determine how the network is trained(Eg: Learning Rate). In machine learning, hyper parameter optimization or tuning is basically the problem of choosing a set of optimal hyper parameters for a learning algorithm. The value of these parameters are used to control the learning process. From Figure 8, We can see

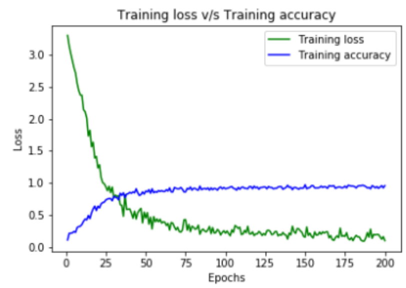

Fig. 8. Loss and Accuracy with Epoch

that the training loss is reducing when the number of epochs approaches to a particular value and the training accurary keep on increasing at starting and remains constant with out much fall when the epochs are in between 100 and 200. So we had chosen a value for epochs by considering the results of these all experiments.

2) Dropout Analysis: Dropout can be defined as a technique that we generally use to prevent a model from overfitting. Basically this Dropout concept works by randomly setting the outgoing edges of hidden units to 0 at each update of the training phase. Here the model is trained with and with out drop out layer and results were analyzed. It is shown in the Fig 9. The accuracy obtained on the testing set is not much very different than the one obtained from the model without dropout. This is due to the limited number of samples that considered here. The details regarding the accuracy are shown in the Table II.

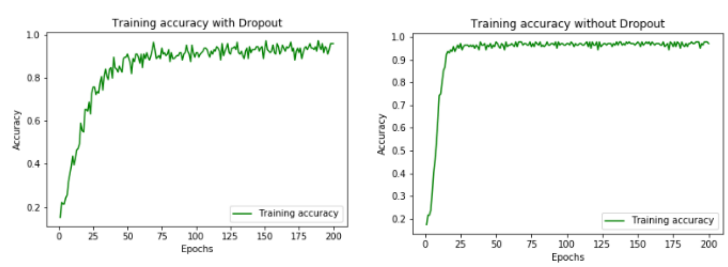

Fig. 9. Network with and with out Dropout

TABLE V
DROPOUT ANALYSIS

Accuracy(%) with dropout	93
Accuracy(%) without dropout	95

B. Analysis

From the experiments that had performed, we had an idea regarding how the accuracy of the system can be improved. The change in accuracy with difference in number of epochs and also by adding and removing the dropout layers are considered and based on that we end up in the current model. For evaluating the performance of the implemented retrieval based system, precision at n concept were used and system generates an accuracy of about 90% in the test. The experiment was done manually by asking random queries. Out of the 20 questions asked 18 were answered correctly. When the number of questions increased also, the accuracy remains like that. It is noted that if we again increase the size of the dataset by adding new queries, we can even improves the accuracy.

VII. CONCLUSION AND FUTURE SCOPE

Conversation AI is one of the growing research are in the field of Artificial Intelligence. The conversational engine can be used in any of the field for providing needed customer support. It is widely used in Educational Websites for querying about the Institutional Details, Medical field for taking doctor appointments, Business field for customer support etc. The proposed system is a retrieval based chatbot which works based on nueral network. Deep Learning concept is used for the model building. As it is a retrieval based system, it particularly works in a closed domain. Here we constructed a Dataset of Bus ticket Booking Query and the system works well on that domain The Training accuracy of the proposed system is 93%. Here when a user ask any query within the domain, the system will generate the corresponding response for it and drive the conversation in a meaningful way. Manual updation of the dataset is needed at regular intervals for improving the efficiency of the system and to help the system to answer most of the repeatedly asked queries of the user. Special features are also added to the model in-order to make the system more user friendly.

Apart from the current model, more improvements can be further added in order to increase the efficiency of the system. Some of the possible modifications or areas of improvements that can be done are mentioned here. Here for developing the model Keras NN is used, so there is a lot of room for improvement. Other type of nueral networks such as Convolutional Network(CN) or Reccurant Network(RN) can also be used to make the system more efficient if needed and not only that Apart from keras framework, There are many more deep learning frameworks such as tensorflow, Apache Spark, PyTorch, Sonnet, and more. Many of the studies shows that there will be higher efficiency for these frameworks when we compare it with that of keras. The next improvement that can further be made is to convert the current retrieval based system in to a combination of both retrieval and generative model. So instead of randomly picking the responses from repository, the system can be made to produce answers by in-coorperating ideas of Natural Language Generation.

REFERENCES

[1] J. Singh, M. H. Joesph, and K. B. A. Jabbar, "Rule-based chabot for studentenquiries," in Journal of Physics: Conference Series, vol. 1228, p. 012060, IOPPublishing, 2019

[2] O. Vinyals and Q. Le, "A neural conversational model,"arXiv preprint-arXiv:1506.05869, 2015

[3] Y. Wu, Z. Li, W. Wu, and M. Zhou, "Response selection with topic clues forretrieval-based chatbots,"Neurocomputing, vol. 316, pp. 251–261, 2018

[4] B. Hancock, A. Bordes, P.-E. Mazare, and J. Weston, "Learning from dialogueafter deployment: Feed yourself, chatbot!,"arXiv preprint arXiv:1901.05415,2019

[5] P. Lison, "Probabilistic dialogue models with prior domain knowledge," inPro-ceedings of the 13th Annual Meeting of the Special Interest Group on Discourseand Dialogue, pp. 179–188, Association for Computational Linguistics, 2012

[6] O. Dusek and F. Jurccek, "A context-aware natural language generator for di-alogue systems,"arXiv preprint arXiv:1608.07076, 2016.

[7] J. D. Williams and G. Zweig, "End-to-end lstm-based dialog control optimizedwith supervised and reinforcement learning,"arXiv preprint arXiv:1606.01269,2016.

[8] R. Yan, ""chitty-chitty-chat bot": Deep learning for conversational ai.," in IJCAI, vol. 18, pp. 5520–5526, 2018.

[9] A. V azquez, D. Pinto, and D. Vilari no, "A computational model for automaticgeneration of domain-specific dialogues using machine learning," inProceedingsof the XVIII International Conference on Human Computer Interaction, pp. 1–2,2017.

[10] B. Wu, B. Wang, and H. Xue, "Ranking responses oriented to conversationalrelevance in chat-bots," inProceedings of COLING 2016, the 26th InternationalConference on Computational Linguistics: Technical Papers, pp. 652–662, 2016

[11] S. Konstantopoulos, "An embodied dialogue system with personality and emo-tions," inProceedings of the 2010 Workshop on Companionable Dialogue Sys-tems, pp. 31–36, Association for Computational Linguistics, 2010.

[12] E. Varghese and M. R. Pillai, "A standalone generative conversational interfaceusing deep learning," in2018 Second International Conference on Inventive Com-munication and Computational Technologies (ICICCT), pp. 1915–1920, IEEE,2018

[13] B. Salehi, P. Cook, and T. Baldwin, "A word embedding approach to predict-ing the compositionality of multiword expressions," inProceedings of the 2015Conference of the North American Chapter of the Association for ComputationalLinguistics: Human Language Technologies, pp. 977–983, 2015

[14] N. Shakhovska, O. Basystiuk, and K. Shakhovska, "Development of the speech-to-text chatbot interface based on google api," inCEUR Workshop Proceedings,vol. 2386, pp. 212–221, 2019