iJRASET
International Journal For Research in
Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 9 **Issue:** VI **Month of publication:** June 2021

DOI: https://doi.org/10.22214/ijraset.2021.35066

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.429
Volume 9 Issue VI June 2021- Available at www.ijraset.com

J.A.R.V.I.S Organizational Virtual Assistant

Prof Vivek Nagargoje[1], Aniket Jadhav[2], Aditya Ghodekar[3], Pranav Chinchwade[4], Maroof Khan[5]
[1, 2, 3, 4, 5]Dept of information technology PCET'S NMIET (SPPU), Talegaon[Dh], India

Abstract: Chatbots, or conversational interfaces as they are also known, present a new way for individuals to interact with computer systems. Traditionally, to get a question answered by a software program involved using a search engine, or filling out a form. A chatbot allows a user to simply ask questions in the same manner that they would address a human. The most well known chatbots currently are voice chatbots: Alexa and Siri. However, chatbots are currently being adopted at a high rate on computer chat platforms. The technology at the core of the rise of the chatbot is natural language processing ("NLP"). A simple chatbot can be created by loading an FAQ (frequently asked questions) into chatbot software. The functionality of the chatbot can be improved by integrating it into the organization's enterprise software, allowing more personal questions to be answered, like"When is the meet?", or "What is the schedule of my day?". A chatbot can be used as an "assistant" to a live agent, increasing the agent's efficiency.

Index Terms: chatbot, Natural Language Processing, NLP, Organization's enterprise software

I. INTRODUCTION

A chatbot, also known as a conversational agent, is a computer software capable of taking a natural language input and providing a conversational output in real time [1]. This human-chatbot interaction is typically carried out through a graphical user interface based on humancomputer interaction (HCI) principles [2], [3]. The idea of an intelligent machine engaging in human interactions was first theorized by Alan Turing in 1950 [4], [5]. Shortly after, automated computer programs, referred to as "bots", were created to simulate human conversation. For example, ELIZA in 1966 matched user prompts to scripted responses, and Artificial Linguistic Internet Computer Entity (ALICE) in 1995 introduced natural language processing (NLP) to interpret user input[4]. Chatbots now exist in various messaging platforms, such as Facebook Messenger, Skype, and Kik, largely for customer service purposes[6]. Chatbots also evolved to interact via voice as well. Such chatbots are typically known as virtual assistants. In particular, the use of NLP led to the Big Four Voice AssistantsApple's Siri (released as a standalone app in 2010, bundled into iOS in 2011, and added to the HomePod device in December 2017), Microsoft's Cortana (2013), Amazon's Alexa (released with its Echo products in 2014).

As evident, chatbots have become quite popular over the years. This is likely due to the rise of Internet users worldwide – there were 3.15 billion users in 2015, 3.39 billion in 2016, and 3.58 billion in 2017 [11]. There has also been a rise in e-commerce, as shown in Fig. 1, coupled with an increased demand for customer service on digital platforms [12]. According to Harvard Business Review, a mere five-minute delay could decrease a business's chances of selling to a customer. In fact, a tenminute delay could reduce their chances by 400% [13]. This paper focuses on creating an AI implemented chat-bot which acts like a personal assistant to set a user's meeting with his colleagues or friends through email conversations. It reads the appointment request email from the client with the help of its natural language processing algorithms like pattern matching. From the info obtained, it checks user's availability at the given date & time from user's google calendar. Accordingly, it generates a reply by natural language generation to send it to user. Through several such interactions and correspondence emails it finally fixes a meeting and makes its entry in user's google calendar. It can also initiate a meeting request and follow the same procedure later.

However, a biased view of gender is revealed, as most of the chatbots perform tasks that echo historically feminine roles and articulate these features with stereotypical behaviors. Accordingly, general or specialized chatbots automate work that is coded as female, given that they mainly operate in service or assistance related contexts, acting as personal assistants or secretaries.

Soon we will live in a world where conversational partners will be humans or chatbots, and in many cases, we will not know and will not care what our conversational partner will be.

II. LITERATURE SURVEY

Naeun Lee et al. [2] [2017] proposed the implementation of word segmentation using NLTK. Natural Language ToolKit (NLTK) is a python package which caters to provide services for NLP. It has inbuilt tokenizers. Users need to import the package and use the required type of tokenizer which is present in the form of functions. The NLTK includes a wide range of tokenizers which are as follows standard, letter, word, classic, lowercase, N-gram, pattern, keyword, path, etc. The most commonly used tokenizer is the word-punkt tokenizer which splits the sentences at the blank spaces.

The accuracy, speed and efficiency of the NLTK tokenizers is commendable. Also, it does not require any algorithm implementation as the package executes them at the backend.

In Bo Chen [6] [2011] proposed a method for implementing the dependency tree. It initially finds out the dependencies among the words in the sentence. Each word is checked for its relationship or dependency with the other word. The word with the highest dependency is selected to be the root. The other words with a relation with the root node are attached to it as the child nodes. This keeps on continuing until all the words are placed in the tree. The tree form of the sentence is called the dependency parser tree. The dependencies among the words are found out by using the POS tags.

LinHua Gao et al. [8] [2018] explains the traditional dictionary method of synonym extractions. In this method, the system database maintains a dataset of synonyms for important keywords in that domain. The sentence sent by the user is then mapped on to that synonym dataset. The keywords detected from the sentence are then checked in that synonym set to check for same intent. All possible synonyms of that keyword are then looked out for a match in the main database. The sentence which is closest to the user sentence is extracted. This method is time consuming and requires more of storage and complexity.

Sijun Qin [9] [2015] proposed a feature selection method for synonym extraction. In this method, among all the parts of speech tags, words having the tags as noun, verbs and adjectives are marked as positive tags and the others as negative tags. The polarity for each feature (word) is then carried out by using the POS tags. If the overall feature polarity is positive, then it can be identified categorically. All the positive features are then grouped together and the synonyms detection for the group of features will be relatively strong, as an entire clause is checked for its synonymic meaning. The synonym sets which are extracted for that clause of features is then calculated for information gain. The one with the highest information gain is the strongest synonym extracted.

Sachin S. Gavankar et al. [10] [2017] proposed the eager decision tree algorithm for prediction. This type of decision tree is the improvised version of the traditional decision tree. It creates this tree at runtime, based on the user"s queries and keeps updating the tree on new user messages. Consider its working for disease prediction. In this algorithm, the symptoms detected in the user query are added as child nodes to the root node. The nodes keep on getting added for new symptoms detected. Further for every symptom, the algorithm checks for the second symptom which has the highest occurrence with the earlier symptom and asks the user for that symptom. If he says yes, then the system traces that path to check for the disease present at the root node. This will keep iterating for all users and the tree keeps getting updated for new entries or traces the path available..

In Liner Yang et al. [5] [2018] put forth the technique of implementing the POS Tagger using Neural Networks. This algorithm consists of „n" numbers of hidden layers. Theses layers are determined by the number of iterations or combinations required to tag the required sentence correctly. At each layer of the algorithm, each word in the sentence is tagged with an appropriate POS tag and then passed to the next later for checking the correctness of the tags. This keeps happening unless the next layer provides the same tags as provided by the previous layer. Another technique to implement the POS tagger is following the traditional approach i.e. of maintaining a dictionary of tags for the given language. Python NLTK provides an inbuilt Tagger which can be used just by importing the NLTK package. The NLTK has a predefined set of tags and a trained data of its own. It tests the sentence and applies an appropriate tag to it. On comparing the above three algorithms, the NLTK tagger proves to be speed and usage efficient. But highest accuracy is provided by the neural network algorithm as it undergoes many iterations.

In Sijun Qin [9] [2015] proposed a feature selection method for synonym extraction. In this method, among all the parts of speech tags, words having the tags as noun, verbs and adjectives are marked as positive tags and the others as negative tags. The polarity for each feature (word) is then carried out by using the POS tags. If the overall feature polarity is positive, then it can be identified categorically

III. THE PROPOSED METHOD

We propose a system which will work as an application and give users information about different kinds of organizational related queries. This application will work using a pattern matching algorithm using depth first search (DFS). In this project, our responsibilities included reading the user inputs and then respond to the query, while trying to keep the conversation related to company environment. The first step in developing the FAQ bot consisted of extensive brainstorming and writing down as many questions as possible. This assisted in allowing FAQ bot to intelligently match pattern (inputs). For doing that we created new AIML files and coupled it with the conversational knowledge base of ALICE bot.

In this module, we are performing some basic operationson the human face to get the proper image for processing.In this module, we perform certain operations like grey-scale conversion, smoothing, edging and image segmentation to get a proper and clean image. In this module, we are performing an algorithm on the human face like eyes detection algorithm to detect the eyes pattern for sleepiness detection.

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.429
Volume 9 Issue VI June 2021- Available at www.ijraset.com

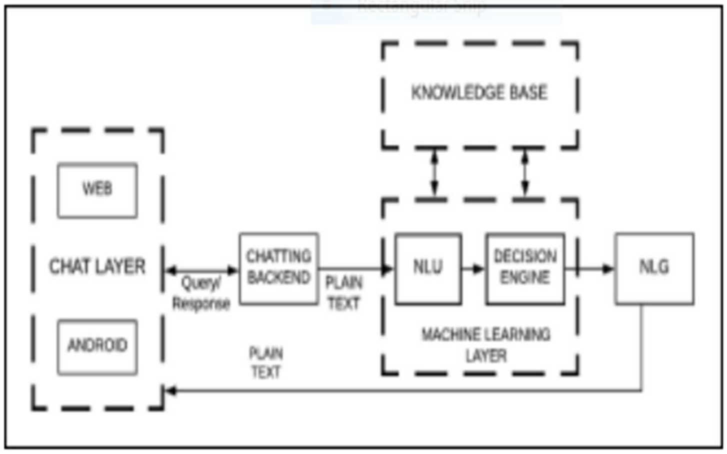

Fig. 1. System Overview

Proposed system uses gated end-to-end memory networks model. This is an end-to-end supervised model with an automatically learned gating mechanism to perform dynamic regulation of memory interaction [11]. This will replace hard attention mechanism in Memory networks where attention values are obtained by softmax function. Gated End-to-End Memory network uses the idea of adaptive gating mechanism of Highway Networks and integrate it into MemN2N. Highway Networks, first Highway Networks, first introduced by Srivastava [3], include a transform gate T and a carry gate C, allowing the network to learn how much information it should transform or carry to form the input to the next layer..

Datasets for training, validation and testing are generated by randomly running scripts. These three are mutually exclusive datasets. Validation dataset is used to minimize overfitting as training dataset actually produces an increase in accuracy. A user request implicitly forms a query that contains the required fields for API call. The bot must ask questions for filling the missing fields and eventually generate the correct corresponding API call. The bot asks questions in a deterministic order. Users then ask to update their requests between 1 and 3 times except for specialization. The order in which fields updated is random.

The bot can confirm with users whether they are done with the updates and issue the updated API call. Token number is generated for each appointment. Users always accept the token number and conversation ends

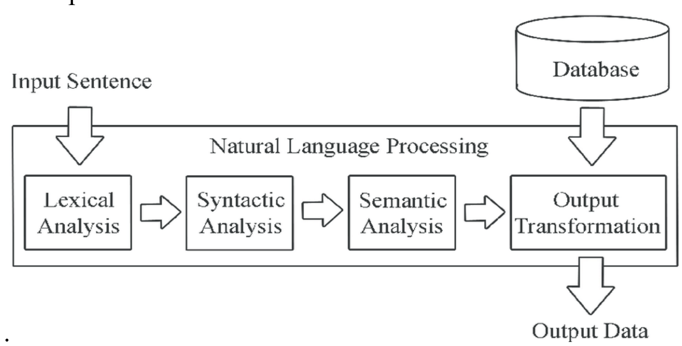

Fig. 2. NLP Working

IV. CONCLUSION

Using pattern matching algorithm, a system that can act as a virtual personal assistant to plan user’s work and schedule his meetings was successfully designed. In terms of the efficiency of the system to respond within a stipulated time period, which achieved overall 70% efficiency, it can be concluded that the system is capable enough to be implemented in the practical world..

Furthermore, the system can be enhanced by including various modules. An important module that can be incorporated is setting the priority of client with whom the user should set meeting, in case of clash of request by multiple clients. This could either be done by getting information about user’s personal relationship with the client through machine learning tools and giving priority to frequent colleagues, or by simply asking priority of client to the user. One of the major part in the system can be enriched is adding the functionality of scheduling meeting with multiple clients for race condition like same time of meeting. Also, the location factor has to be included so as to allocate a meeting place appropriate for both the user as well as the client.

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.429
Volume 9 Issue VI June 2021- Available at www.ijraset.com

REFERENCES

[1] Donna Interactive Chat-bot acting as a Personal Assistan, Namita Mhatre Karan Motani Maitri Shah
[2] Cyril Joe Baby, Faizan Ayyub Khan, Swathi J. N., “Home Automation using IOT and a Chatbot using Natural Language Processing”,
[3] Oliver Pietquin, Thierry Dutoit, “Dynamic Bayesian Networks for NLU Simulation with Applications to Dialog Optimal Strategy Learning.
[4] A. Miller, A. Fisch, J. Dodge, A. Karimi, A. Bordes, and J. Weston. 2016. Key-value memory networks for directly reading documents. In Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing
[5] F. Liu, J. Perez (2017). Gated End-to-End Memory Networks. In Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics,
[6] H. Wang, Z. Lu, H. Li, and E. Chen, (2013). A dataset for research on short-text conversations. In EMNLP.
[7] [Z. Wang, and O. Lemon, (2013). A simple and generic belief tracking mechanism for the dialog state tracking challenge: On the believability of observed information. In Proceedings of the SIGDIAL 2013 Conference.
[8] S. Young, M. Gasic, B. Thomson, and J. D. Williams, (2013). Pomdp-based statistical spoken dialog systems
[9] Kyo-Joong, DongKun Lee, ByungSoo Ko, Ho-Jin, Choi, “A Chatbot for Psychiatric Counseling in Mental Healthcare Service Based on Emotional Dialogue Analysis and Sentence Generation”, IEEE 18th International Conference on Mobile Data Management, 2017
[10] Sijun Qin, Jia Song, Pengzhou Zang, Yue Tan, “Feature Selection for Text Classification Based on Parts-Of-Speech Filter and Synonym Merge”, 12th International Conference on Fuzzy Systems and Knowledge Discover (FSKD), 2015