Intelligent Assistance for Organisation

UMA MAHESWARI B[1], KAMALESH KUMAR. MG[2] , KAUSHIK. S[3]

Associate Professor, St. Joseph's College of Engineering, OMR, Chennai-600 119 INDIA[1]
B.E., St. Joseph's College of Engineering, OMR, Chennai-600 119 INDIA[2]
B.E., St. Joseph's College of Engineering, OMR, Chennai-600 119 INDIA[3]

Abstract **-Today, organizations have multiple tools and systems to help them assist in their day-to-day development activities. Managing these many applications has become a tiring job for the employees as they feel pressure of mental overload in daily tasks that challenge their concentration. So there is a need for an intelligent system (application) which can connect to different applications, mine & analyse data and provide results to employees whenever they need it. The Intelligent system application is hosted in the organization network with complete access to organization resources. The application uses speech recognition techniques to convert employee voice to text. To get the intent of the request natural language understanding technique is used. To support this, the application is pre-trained with conversational data using convolutional neural networks. Based on the intent, data is mined from respective resources by making an API call. The mined data is then analysed; intent action is performed and notified to the employee through voice response. The primary feature of this application is to learn-mine-analyse-serve information to employee queries. The additional feature of this application is to record the meeting, store it in detail in the database and generate summarization reports. The performance and efficiency of this intelligent application is ensured by performing various tests**

Keywords **- voice assistant, intelligent system, natural language understanding, and organization application.**

I. INTRODUCTION

The world today is moving towards automation and artificial intelligence. We all know artificial intelligence (AI) is the future of the world. Chabot and voice assistant is one such artificial intelligence product that can interact and connect with people. These intelligent products have already started to add value and improve the user experience. Text was the main way to get assistance, but now there is an evolution of technology from text to voice based assistance system. The voice based intelligent systems can understand natural language and perform tasks or services for an individual. Siri by Apple, Google home by Google, Alexa by Amazon, Microsoft's Cortana on Windows are some of the voice based intelligent products. But there is no system that offers assistance in corporate organization. As organizations have multiple tools and systems used in day to day software development activities, it becomes a tiring job for an individual to manage these many applications. The employee may feel pressured and mental overload in daily tasks challenging their concentration. So just like how Siri provides assistance for smartphone tasks, the employees in corporate are also in need of a voice based intelligent system that can assist them in their daily work. The intelligent personal assistance can connect to different applications, mine & analyze data and provide the results to users whenever they need it.

The objective is to create an AI/ML based application to be hosted within a corporate/organization intranet with complete access to corporate resources like JIRA, Confluence, Bitbucket, test link, stack overflow, mail etc. The application has to learn, mine, analyze and serve information about a product to optimize development and support processes in a product team. The application also has to get voice input from the user and process it and has to generate voice response or mail response to the user.

II. RELATED WORK

Voice assistants use speech recognition and natural language understanding for processing requests. Initially classification of speech recognition systems and voice recognition technology was performed [1]. Then for NLU exploit of hierarchical Bayesian interpretation for language modeling is done [2]. Then with the rise of the heat of Artificial Intelligence, deep learning has also begun to be popularized in people's daily life [4]. With that research on natural language processing started with so-called rule-based methodology [5]. To achieve this, the experiment implements sequence to sequence long short-term memory cell neural network were used [7]. Later chatbots applications are used in dialog systems for several practical purposes such as customer service and information acquisition [8]. At present, there are personal assistants like Google assistant and Siri that some individuals use for their official work. These assistants can be used for some simple actions like scheduling a meeting, calling someone, setting a reminder, etc., but there is no true corporate assistant to help employees and organization of their work. The current AI assistants cannot connect with different applications nor mine data from them.

III. PROPOSED SYSTEM

The proposed Cooperate assistant is designed specifically for corporate employees to assist or help them in their day to day office work. For example keeping track of a project's progress, get information from different applications without actually opening them, record and auto generate meeting reports. Using a single application or voice assistant one can manage all the other applications that the company uses. The AI voice assistant is hosted on the organization or corporate network. Employees of the corporation can connect with the application, ask queries and get responses.

A. Design and Architecture of AI Personal Assistant

The objective of this project is to create an AI based application to be hosted within a corporate/organization intranet with complete access to corporate resources like JIRA, Confluence, Bitbucket, test link, stack overflow, mail etc. The AI based application acts as an intermediate between the user and the service providing software or application. The intelligent system communicates with both the user and the service providing apps. The intelligent system interacts via voice with the user and interacts via api call with the service provider.

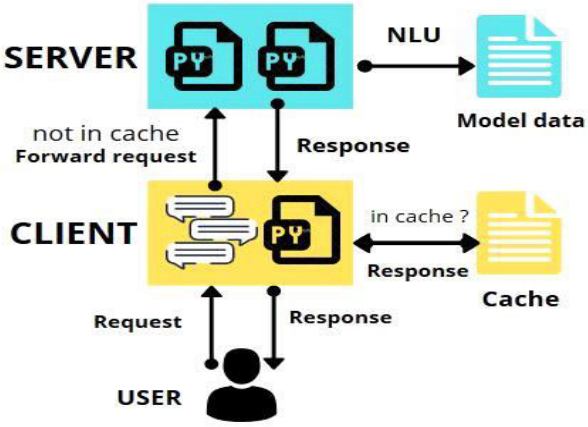


Fig. 1 Application System Architecture

The Fig.1 is a client server architecture diagram. Client Server architecture is the best suited and simple architecture for this project. The Server here is the Artificial Intelligence server. The Client system is connected with the server. Socket binding is used to connect the server and client together. The server listens to the socket on the network. The client using the hostname and port number of the server makes a connection. On server side threading is used to handle client multiple connections. User requests are handled at both client and server side based on the type of request. End user cannot directly request the server. A client side script at each end system enables users to get connection with the server and serve responses. The best part of this architecture is the Client side caching. This caching enables faster and better results. Instead of processing each and every request at the server, some requests can be handled at the client side itself.

B. Processing End Points of system

The server contains two scripts. One script for training the server with conversation json data. Another script for learning from the trained model and process the new request. A json big data is passed as an input to the training script and is trained using naïve bayes classifier. The results of the training script are stored as a model metadata file. The results from these metadata are used while processing requests. So no training needs to be done while processing requests. Processing of requests is done from a pre-trained Metadata model. The server core script processes the user request. The Natural language understanding is performed on the trained model data. Also the server script is responsible for mining data from the other services. So processing, mining, socket binding and response generation all is handled by one single server script. Training script is alone separated from the main server . The client side functionality includes establishing socket connection, speech recognition, request forwarding, caching response etc. The core part of the intelligence system is the server. All new requests are handled by the server. But as the number of clients in the network increases or number of repeatable requests to the server increases, server loading occurs. So it is significant to reduce the load on the server. So client side request handling and caching response is done. The client side contains the user interface, a chat based popup interface to display results to users and speech recognition module, to convert voice to text before processing requests. The client side also converts the text to voice for output as voice. Processing of requests is handled at both server and client side but the enforcement of the response is handled at client side of the architecture.

IV. METHODOLOGY

Every individual person can have their own customizable personal assistance. This AI system is developed in python as it is the most flexible and easier programming language to build machine learning models. The AI system accepts voice input of the user. To get voice input from the user Google speech API is used. It also converts the voice input into text which can be fed to the machine learning module. After obtaining the text from the voice input, the intent of the text needs to be determined. For that naïve bayes algorithm for text classification and natural language processing is used and user intention is determined. After knowing the user intention next is to make an API call to the corresponding application and mine data. As most of the applications used in corporations have their own API's for development purpose, it is easy to mine data. For each operation that is to be performed on the application, there are inbuilt functions in python to make a call to the API to obtain data. The API provides the data in json format. The AI System then parses through the json data to obtain relevant information of user queries. For example, if a user gives a voice input to get repositories from bitbucket. The voice input is first converted to text and then the naïve bays NLP processes this text and determines the intention. Now the AI system will make a call get_repositories () to the bitbucket API. Bitbucket returns a json data which contains details of repositories. The AI system analyses the json data and gives the details of the repository to the user. The user can even ask the AI system to open the repository in the browser. Speech recognition and response enforcement is done on client side script. Request processing, mining and training is done on server side script. Request handling is done at both server and client. Client handles repeated requests which are already cached, while server handle new requests.

A. Speech recognition module at client end

The speech recognition module is at the client side of the architecture. The voice input of the user is converted into text using this module. Google speech recognition api, commonly used api for voice to text convection is used for this module. The speech recognition systems rely on the Hidden Markov Model. The Speech signal is split into 10 ms fragments which is then mapped to real number coefficients. The output of the Hidden Markov Model or HMM is a sequence of vectors. Groups of these vectors are matched to one or more fundamental units of speech to decode speech to text. To support speech recognition, a pyaudio library is used. The pyaudio package is used for capturing microphone input. Recognizer class in speech recognition is used to recognize the speech. Also there is another method record() to voice.

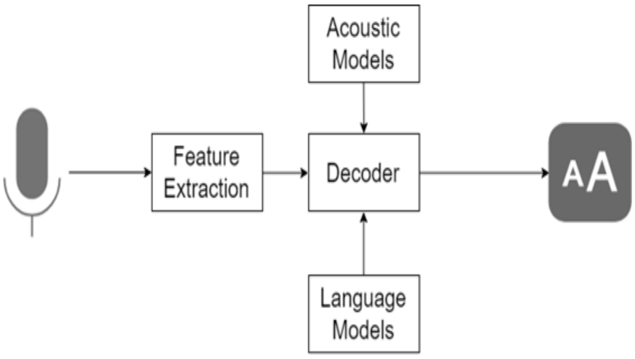


Fig. 2 Speech Recognition Module Representation

B. Natural Language Understanding (NLU) module

The Natural Language Processing (NLP) enables intelligent systems to understand the user requests. But for being intelligent system to provide personalized conversation and user experience, the conversation engine unit in NLP helps. The major aspect of this intelligent system conversation engine is intent classification. A classifier is a method of categorizing data , in this case user requests into many different categories. Much alike how humans will classify objects into sets As simple as it may sound, it is actually quite a complex process. User request input is identified by classifier function, and associates the information with a specific "intent", with a detailed explanation of the words for the AI system to understand. To perform Natural Language Understanding (NLU), intent classification needs to be done . The Intelligent system has to be trained to respond for a particular query. The training data must be as simple, so that the model can train faster and easier. The raw data should be cleaned before feeding into the model. This can be done by many methods. Following the Lemmatization process, another technique to proces or group different forms of a word is done. Next step is to provide some training data. Each word from the given sentence is tokenized, stemmed, and lowercase, this is consistent with the transforms we applied to the corpus data.

1. Naive Bayes for Natural Language Understanding

Naïve Bayes is a statistical algorithm that is used for classifying text. Bayes' Theorem is based on the probability of an occurring event when given the probability of another event that has already occurred. Naive Bayes is a family of algorithms which share a common principle where. all pairs of classified features are independent of each other. The Multinomial Naive Bayes is used for breaking down sentences into words (tokenization): “hello children”into a list of individual words or tokens: “hello”, children”. It reduces words to their stem (stemming): “children” stems to “child”. The collection of all stemmed words is called “corpus”. For example the stem of the word “have” is “hav” so that it matches the stems for “having”.

C. Data mining module of AI Server

The data mining module of the project is where the intelligent system makes API requests to the service providing software. Most of this software provides Restful services. The Service REST APIs are used to interact with the Service Server applications remotely. The workflow features provided by the services using REST API. The authentication for this REST API is done using OAuth. Service REST APIs provide access to resources (that is, data entities) via URI paths. To use a REST API, your application makes an HTTP request and parses the response. The service REST API uses JSON as its communication format and the standard HTTP methods like GET, PUT, POST, and DELETE. Resource expansion is used by the service REST API is used to simplify the process, which is nothing but to return explicitly only the requested parts of the resource. This helps to avoid problems that occur when only little information is requested. Also there is a specific package in python for famous services like Atlassain-api-package for jira and confluence because they all are developed by the same company. API requests to the service are made by requests package in python. The requests package is a Python HTTP library with Apache License. The aim of this model is to make HTTP request. The package has many features for parameter, custom headers etc.

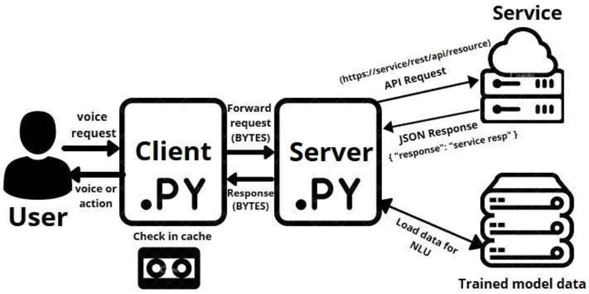


Fig. 3 Server Connection for Different Services

D. Data Analysis of API response

JSON standard made transportation of data between client and server easier. The service provider gives a Json response to the intelligent system's REST API request. JSON is meant to be easily readable by humans but it has been a bit tricky to work with JSON data in general. The JSON data is nested and hierarchical. The attributes of the data have several key-value pairs. Python has a built-in package named json for encoding and decoding JSON data. Import json library package in python. Serialization is the process of encoding JSON This term refers to the transformation of data into a series of bytes to be stored or transmitted across a network. Deserialization is nothing but decoding the JSON standard data. Here the service provider JSON response is deserialized. In the json package, load() and loads() is used to convert JSON encoded data into Python objects.

```
parsed_json = (json.loads(json_data))
```

The JSON is like a string unless it is obtained from a file. This can be saved to a variable and iterated.

```
myRecords = resp.json()["sections"]
myRecords = resp.json()
df = pd.read_json(json.dumps(myRecords), orient='records')
```

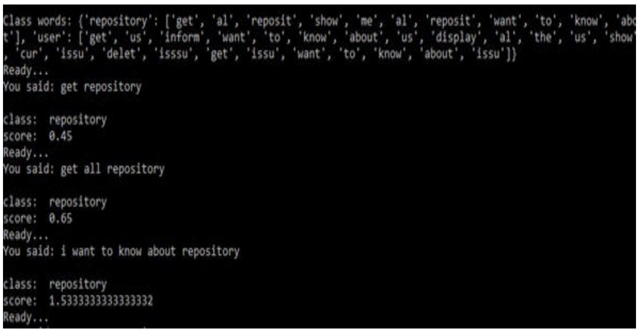
Parsing JSON in Python is not really hard. One can convert the JSON into a dictionary by using the json.load method. The dictionary can also be imported into an object as it's instantiated to transfer data into a new object.

[{"expand":"description,lead,url,projectKeys","self":"http://localhost:8080/rest/api/2/project/10000","id":"10000","key":"DEMO","name":"demo","avatarUrls":{"48x48":"http://localhost:8080/secure/projectavatar?avatarId=10324","24x24":"http://localhost:8080/secure/projectavatar?size=small&avatarId=10324","16x16":"http://localhost:8080/secure/projectavatar?size=xsmall&avatarId=10324","32x32":"http://localhost:8080/secure/projectavatar?size=medium&avatarId=10324"},"projectTypeKey":"software"}]

The above raw data is a json response for get_projects() in jira. The response contains details of projects. The intelligent system will parse this json response and extract required data and provide it to the end user.

V. EXPERIMENTAL RESULTS

As a developer, testing of software is done while building. The testing of intelligent voice systems is not very hard, but pretty much straightforward. An Intelligent system works in a way where the user gets response for their requests. The NLP module which the AI system uses processes the request which is provided by the user and gives a relevant reply. Creating a conversational dataset from which the AI system trains to give the relevant reply is the job of the developer. This testing enables to check if the bot provides the correct answer to the question. The AI System should be able to process any kind of requests which the user types. Testing this feature involves technical aspects while developing the NLP model. The NLP model should be made in such a way that the bot is able to understand any kind of input, which is irrelevant to the system's main functionality.



```
Class words: {'repository': ['get', 'al', 'reposit', 'show', 'me', 'al', 'reposit', 'want', 'to', 'know', 'abo
t'], 'user': ['get', 'us', 'inform', 'want', 'to', 'know', 'about', 'us', 'display', 'al', 'the', 'us', 'show'
, 'cur', 'issu', 'delet', 'isssu', 'get', 'issu', 'want', 'to', 'know', 'about', 'issu']}
Ready...
You said: get repository

class:  repository
score:  0.45
Ready...
You said: get all repository

class:  repository
score:  0.65
Ready...
You said: i want to know about repository

class:  repository
score:  1.5333333333333332
Ready...
```

Fig. 4 Result of Natural Language Understanding

Using naïve bayes classifiers, each request is mapped to a class with a score. Higher the score, better understanding of the request by server. Another important factor is the speed at which the bot replies to the messages. If a Bot takes a long time to reply, a time like a person takes to reply an email, what is the point of having a bot for the particular purpose. Hence, the speed of the bot should be tested while developing and also by getting beta users on board.

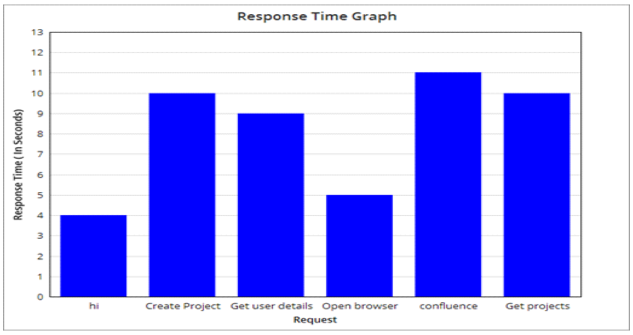


Fig. 5 – Response Time Graph

AI systems we build are meant to reply utilizing the predefined utterances. Out of the predicted utterances, the number of utterances which the bot gets correct is said to be the accuracy of the bot. While developing and testing the bot, it is very important to maintain a high accuracy for the intelligent system.

VI. CONCLUSION

The current work shows a framework for providing assistance to employees in corporations or organizations. As organizations have multiple tools and systems used in day to day software development activities. It becomes a tiring job for an individual to manage these many applications. The employee may feel pressure and mental overload in daily tasks that challenge their concentration. So just like how Siri provides assistance for smartphone tasks. The employees in corporate are also in need of a voice based intelligent system that can assist them in their daily work. The intelligent personal assistance can connect to different applications, mine & analyze data and provide the results to users whenever they need it. By performing various testing, it is guaranteed that the system works fine and the users can react with it in an efficient way. The test results show a promising result.

REFERENCES

[1] Jianliang Meng, Junwei Zhang,Haoquan Zhao: “Overview of the Speech Recognition Technology”, Fourth International Conference on Computational and Information Sciences, 2012, 17-19 Aug. 2012, DOI 10.1109/ICCIS.2012.202

[2] Songfang Huang, Steve Renals: “Hierarchical Bayesian Language Models for Conversational Speech Recognition”, IEEE TRANSACTIONS ON AUDIO, SPEECH, AND LANGUAGE PROCESSING, VOL. 18, NO. 8, NOVEMBER 2010, 19 January 2010, 10.1109/TASL.2010.2040782

[3] R. Yasotha, E.Y.A. Charles: “Automated Text Document Categorization”, 2015 IEEE Seventh International Conference on Intelligent Computing and Information Systems (ICICIS'15), 12-14 Dec. 2015, 10.1109/IntelCIS.2015.7397271

[4] Song Peng, Li Zhijie, Geng Chaoyang: “Research on Text Classification Based on Convolutional Neural Network”, 2019 International Conference on Computer Network, Electronic and Automation (ICCNEA), 27-29 Sept. 2019, 10.1109/ICCNEA.2019.00052

[5] Hitoshi ISAHARA: “Resource-based Natural Language Processing”, 2007 International Conference on Natural Language Processing and Knowledge Engineering, 30 Aug.-1 Sept. 2007, 10.1109/NLPKE.2007.4368002

[6] Zhang Zhaoyin, Li Yanfang: “Rule-based Natural Language Understanding Based on Fuzzy Evaluation of Teaching Quality”, 2009 International Forum on Computer Science-Technology and Applications, 25-27 Dec. 2009, 10.1109/IFCSTA.2009.65

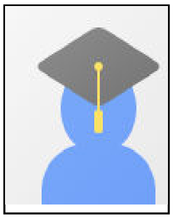
[7] Milla T Mutiwokuziva, Melody W Chanda, Prudence Kadebu, Addlight Mukwazvure, Tatenda T Gotora: “A Neural-network based Chat Bot”, 2017 2nd International Conference on Communication and Electronics Systems (ICCES), 19-20 Oct. 2017, 10.1109/CESYS.2017.8321268

[8] Namrata Bhartiya, Namrata Jangid, Sheetal Jannu, Purvika Shukla, Radhika Chapaneri: “Artificial Neural Network Based University Chatbot System”, 2019 IEEE Bombay Section Signature Conference (IBSSC), 26-28 July 2019, 10.1109/IBSSC47189.2019.8973095

[9] J. Todorov, S. Stoyanov, V. Valkanov, B. Daskalov, I. Popchev: “Learning Intelligent System for Student Assistance - LISSA”, 2016 IEEE 8th International Conference on Intelligent Systems (IS), 4-6 Sept. 2016, 10.1109/IS.2016.7737397

[10] László Horváth, Imre J. Rudas: “Human Interactions for Self-Adaptive Virtual Product Prototype”, 2013 IEEE International Conference on Systems, Man, and Cybernetics, 13-16 Oct. 2013, 10.1109/SMC.2013.797

AUTHORS PROFILE



Dr UMA MAHESWARI B is a M.E. and PhD holder. She is currently working as an Associate Professor in Department of Computer Science, St. Joseph’s College of Engineering, Chennai. She has presented various research papers in international conferences and published research work in different journals. Her areas of expertise include software engineering and testing. She also has mentored many undergraduate and graduate students in their project and research work. At present, she is actively researching in the field of Machine Learning and Data Science.

KAMALESH KUMAR MG has completed his bachelors in computer science and engineering from St. Joseph’s College of Engineering, Chennai. He is currently working as a software developer engineer at Zoho Corp. His areas of interest include Data Science, Machine Learning and Software Engineering. He completed Machine Learning certification offered by Stanford University via Coursera. Also, he participated in various events and symposiums in the field of computer science. In 2019, he led his team to the finals of Smart India Hackathon Software Edition. At present, he is actively exploring the field of machine learning and data science .

KAUSHIK S has completed his bachelors in computer science and engineering from St. Joseph’s College of Engineering, Chennai. He is currently pursuing his masters in Artificial Intelligence and Data Science from Amrita University. He is interested in the fields of data science, machine learning, IOT, and networking. During his under graduation he has worked on various projects. One such work was an IOT project for controlling electronic appliances using smart phones. He also participated in various technical events. He was one of the runners-up of Smart India Hackathon 2019 software edition. At present, he is researching data science as part of his master’s program.