Artificial Intelligence powered Digital Assistant for Voice over Internet Protocol (VoIP)

Raghav Sridhar, Research Scholar, BITS Pilani, sridhar.raghav29@gmail.com

Parthasarathy P.D. , Research Scholar, BITS Pilani, pdparthasarathy.03@gmail.com

Dr. Vinod Vijayakumaran, Professor, BITS Pilani, vinod.vijayakumaran@gmail.com

Abstract: **This research aims to provide a method to connect an intelligent, AI powered digital assistant to any VoIP based service. The advent of virtual assistants such as Apple Siri, Amazon Alexa, Microsoft Cortana etc. into the digital world has made the world an extremely connected place. However, most of these virtual assistants are tied to the offerings provided by the parent company that has invested into their development. In the present market, there doesn't exist a fully open source virtual assistant which can be configured to dynamically react based on a real-world conversation. Most virtual assistants are preprogrammed to do very specific tasks which do not really provide tangible real-world value. We aim to provide a smart solution in the form of a service which can be consumed by VoIP (Voice over Internet Protocol) providers to make their solution more intuitive and eases a user's experience.**

Keywords —Machine Learning, Service Oriented Architecture, VoIP, digital assistant, virtual assistant

I. INTRODUCTION

Modern virtual assistants are a dime a dozen. However, not all of them offer services which can be used by every VoIP (Voice over Internet Protocol) provider and, it is not very easy to integrate them. Leveraging the power of machine learning, digital assistants are mostly used to provide services and actions which are not always business relevant, i.e., the digital assistants consume more than they offer. We propose, in this paper, a way to use the services provided by virtual assistants, in a user-friendly manner. There is a growing concern over security and how data is being used in this era of technology. As the focus of supporting systems is shifting towards holding more data within 'the Cloud', the privacy and security of this data has become a cause for concern. With this area of social concern in mind, the idea of a VPA becomes attractive as it changes the focus of the supporting system to the contextual sphere under private control of the user [1]. This paper will provide a method to introduce a smart virtual assistant – a digital assistant into the world of VoIP. It will aim to eliminate the requirement for members to follow up on meetings and create appointments on their calendars, in a secure and structured manner, using Service Oriented Architecture.

II. CURRENT APPROACH

As of now, most VoIP digital/virtual assistants do not offer an advanced machine learning based approach to handling actions dynamically.

What is generally offered on the market is restricted but not limited to:

- Attending calls and providing routing options
- Answering basic queries by providing number-based options
- Requesting to take voicemails or messages to be added to the inbox
- Relaying pre-recorded messages for providing customers with a professional feel.
- Staying as an active listener in calls, just for the sake of managing meeting rooms that are booked, such as Skype room system.

As we can see, almost all of the aforementioned points are reserved to quality-of-life offerings and do not necessarily offer any real value.

A virtual assistant ensures that your customers or anyone else calling in, are directed to the most appropriate resource. Even if you have a fully staffed call center (or don't), the virtual assistant can save time and improve productivity. Often called auto attendant or digital receptionist, companies use them to project a professional image to callers. There are clear benefits to using virtual assistants. Your trained technicians or customer service representatives don't have to waste time answering simple questions. Their expertise can be put to better use handling more complex issues. It reduces hold times for customers as well. Virtual assistants in phone systems are not new but more businesses now have the option of using them without paying exorbitant prices for the privilege. [2]

None of the present VoIP providers offer any real intuitive service which can make a host's job easier. Even if it is provided, the service is neither cheap nor open source, and very restricted in the usage rules.

 DOI : 10.35291/2454-9150.2019.0041

III. CHALLENGES IN CURRENT APPROACH

Voice-based digital Assistants such as Apple's Siri and Google's Now are currently booming. Yet, despite their promise of being context-aware and adapted to a user's preferences and very distinct needs, truly personal assistants are still missing [3].

The current approach has a few challenges. Some which are very impactful are:

- The current approach is restricted to not performing continuously improving actions, which may result in an additional cost in requisition of resources.
- No proper open source services available to provide such a service.
- Consumes real-time human resource to perform non-technical activities such as book-keeping, which leads to a loss of productive time and also increases the total cost of operations in the long run.

IV. PROPOSED SOLUTION

Voice-based interaction is usually simple, flexible and does not require cognitive efforts, attention and/or memory resources on the side of the user. Voice interfaces for example, can flatten option menus and supply rapidly complex verbal responses [3].

The main purpose of this research is to provide a means to offer digital assistant capabilities over VoIP offerings. Often, in business meetings, a single person takes the minutes of a meeting, and a plan is formulated after the meeting based on the points of note from the text jotted down. This introduces a lot of ambiguity, as the readers are left at the mercy of the person observing the minutes of the meeting. Some issues which are introduced as a result of humans noting the minutes of a meeting:

- Human error is common, and there is a good chance that key points are not noted down.
- Across various nationalities, the interpretations of certain sentences might not be the same, i.e., a sentence of sarcasm might be interpreted literally, and vice versa.
- There is a good probability of not following up on many action items, for a variety of reasons; for example, workload, just missing out on certain action items etc.

The digital assistant proposed will make use of a service-oriented architecture (Service-Oriented Architecture (SOA) is an architectural approach in which applications make use of services available in the network. In this architecture, services are provided to form applications, through a communication call over the internet [4]), by leveraging the power of IBM Watson Speech to Text Service and machine learning. Machine learning is a data analytics technique that teaches computers to do what comes naturally to humans and animals: learn from experience. Machine learning algorithms use computational methods to "learn" information directly from data without relying on a predetermined equation as a model. The algorithms adaptively improve their performance as the number of samples available for learning increases. Deep learning is a specialized form of machine learning [5].

BigML is a powerful Machine Learning service that offers an easy-to-use interface for you to import your data and get predictions out of it. The beauty of the service is such that you do not need a profound knowledge of Machine Learning techniques to get the most out of ML [6].

This solution will aim to:

- Provide scribe and listener services to the host of the meeting.
- Be easy to turn active and inactive, in case of confidential meetings
- Make great quality of life improvements to the existing working patterns followed by people.
- Improve business impact by reducing time spent in performing activities such as scheduling meetings and keeping track of action items.
- Provide a proper log of everything spoken, shared and decided in a meeting for future references.

The digital assistant needs to be truly digital, i.e., it should not require more than a simple button to activate or deactivate it. This digital assistant will make use of a few major software components, including some services available on the internet. The components and their workings are as follows:

LISTENER

- Any competent speech-to-text service such as: IBM Watson Speech to Text Service, Google Cloud Speech-to-Text, Cognitive Services Speech to Text etc., with appropriate wrapper classes written to save the text files locally or on a specified cloud storage that has been pre-configured, should be implemented, for our use case we have IBM Watson Speech to Text Service.
- The APIs will allow for an engine to be turned on and off at will.
- When turned on, the service will be added as participant, aka the "Listener" into a conversation over any VoIP service which can consume the same APIs.
- The consuming VoIP service must enable a toggle button in the UI to turn off if the service is unnecessary.
- If the service is turned off mid-way through a VoIP connection, the text until that point is processed and appropriate action items are created.
- As soon as the conversation begins, the Listener makes note of all the participants and labels them accordingly.
- Every time a participant speaks, or performs an action, the Listener (which makes use of the APIs from the speech-to-text service), notes the action into a log file.

 DOI : 10.35291/2454-9150.2019.0041

- The action might be a screen share, or a speech activation, or a message text.
- The Listener stops listening whenever it is toggled off, or if the conversation comes to an end with the termination of the window.
- After the end of a session, the Listener first creates a folder: "Conversation on <date>" and a text file within the same folder with all the events and actions of the conversation, "1st Participant's name> and N others" in the mentioned repository location.
- The Listener is also capable of saving the media version of the conversation.

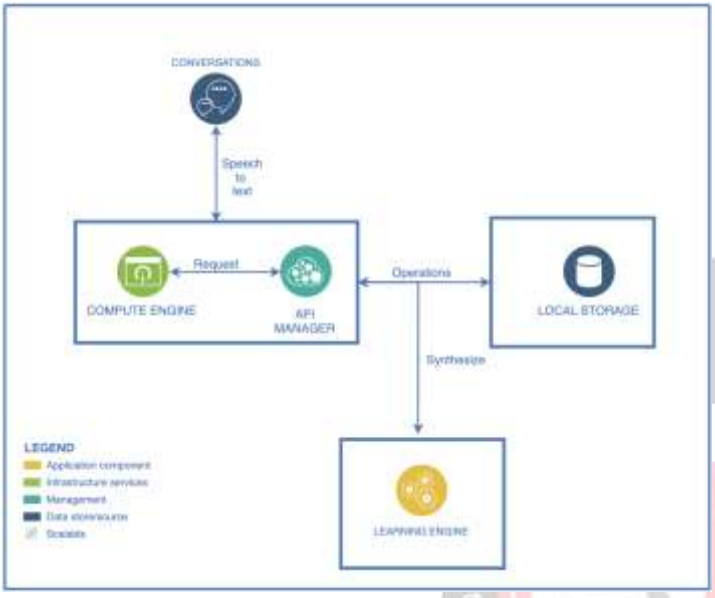

Fig.1 Structural Overview of AI in VoIP

COMPUTE ENGINE

- The Compute Engine is the centralized manager which decides what action to perform after the Listener feeds it the location of the created folder and file.
- This Engine is responsible for routing any specific actions to the API Manager which is a subsection of the same software component (for the sake of better understanding of future scope, it has been separated from the Overview diagram).
- The API Manager contains the extensible services provided by the database services (if any), and the Machine Learning Services.
- It also consists of the wrapper classes written to manage custom operations such as creating a local folder (local repository), writing into any specific repository location, changing machine learning plug-ins, evaluating data sets and segregating them internally into training data and real-time data etc.
- The API Manager is also responsible for firing asynchronous requests to the configured calendar on the VoIP system, to make arrangements for actions which are suggested by the Learning Engine.

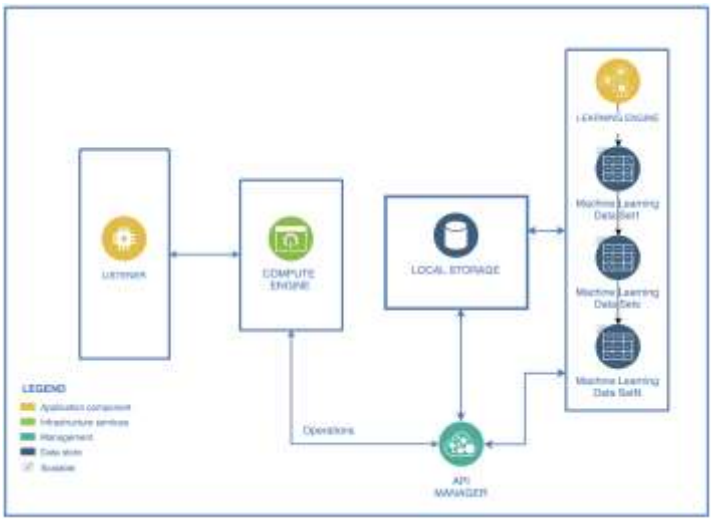

Fig.2 Component View of the service

Local Storage

- This component is responsible for storing the content, primarily locally. It has been configured to provision faster to access to data for all the components, instead of depending on another service to provide the data store facilities.
- However, the local storage can also be configured to point to a remote repository such as AWS S3 Object Store, Google Drive, Microsoft One Drive etc.
- The appropriate adjustments to accommodate operations on the different data stores need to be made accordingly.
- The Local Storage, if maintained on a local system, does not provide archival facilities, and is totally dependent on the user to back up data periodically.
- Files stored on the local store carry a particular naming format which makes it simpler for the engine to access and perform further actions.

Learning Engine

- The Learning Engine is the most vital component of the whole service offering.
- It extends any competent machine learning service available on the internet. For our use case, we have used BigML.
- It takes the data available in the local store, performs data fragmentation and attains data sets.
- As the Overview diagram suggests, the Learning Engine is an application component similar to the Listener, i.e., it is a class which performs certain context-based actions.
- Certain data warehousing algorithms like Periodic Snapshots, Apriori Algorithm etc.
  And creates certain fact tables. These fact tables are interpreted by the Learning Algorithms to provide actionable events for the API Manager to act upon.
  The actionable events might be: CREATE MEETING, DELETE MEETING, UPDATE MEETING TIME, CREATE REMINDER, CREATE APPOINTMENT, UPDATE APPOINTMENT etc.

 DOI : 10.35291/2454-9150.2019.0041

- The API Manager now uses the APIs provided by the VoIP Services to update the calendar of the user accordingly.

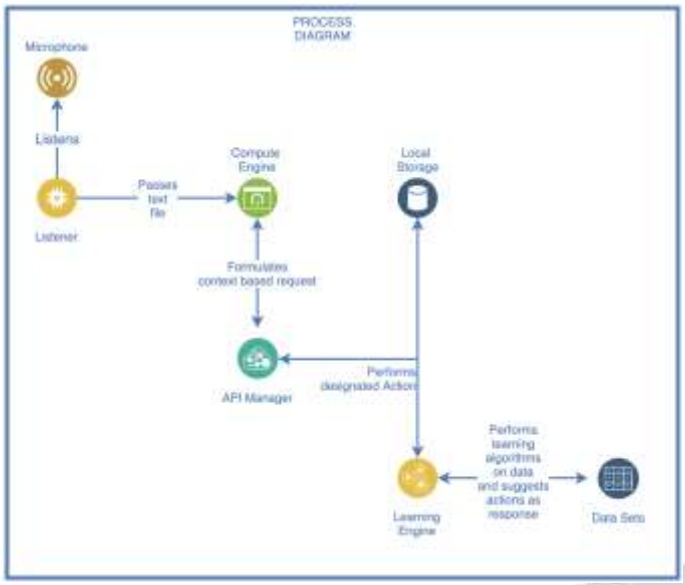

Fig.3 Process View of the service

V. IMPLEMENTATION AND RESULTS

The folder structure that is formed is as follows:

Fig.4 Folder structure created by the service locally

The text is stored primarily into one conversation text file along with the time stamp as shown in Fig. 5.

```
Speaker SRIDHAR,RAGHAV: Hi. [00:00:06]
Speaker SRIDHAR,RAGHAV: <shared "Emoji on messenger: (" :) ")> [00:00:08]
Speaker SRIDHAR,RAGHAV: I will be in the office by six thirty or seven o'clock
in the morning. [00:00:32]
Speaker KUMAR,BHARAT: Hi. [00:00:40]
Speaker KUMAR,BHARAT: Are you doing some work after five in the
evening? [00:00:44]
Speaker KUMAR,BHARAT:. [00:00:45]
Speaker SRIDHAR,RAGHAV: Yes. [00:00:47]
Speaker SRIDHAR,RAGHAV: I would stay back beyond that. [00:00:54]
Speaker KUMAR,BHARAT: Okay right. [00:01:06]
```

Fig.5 Text stored into file

This file is passed as an input to the Learning Engine. The Learning Engine separates each speaker's text into subsequent folders labelled by the user names. Each text file of the user holds the individual lines captured by the Listener as shown in Fig. 6.

The Learning engine creates context-based facts and feeds it as a response in a JSON structure to the Compute Engine. The Compute Engine now fires an event request to the calendar service configured with the VoIP service to create an appropriate event. In this case, as appointment after 5PM as shown in Fig. 7.

```
Speaker KUMAR,BHARAT: Hi. [00:00:40]
Speaker KUMAR,BHARAT: Are you doing some work after five in the
evening? [00:00:44]
Speaker KUMAR,BHARAT:. [00:00:45]
Speaker KUMAR,BHARAT: Okay right. [00:01:06]
```

Fig.6 Individual user text stored into file

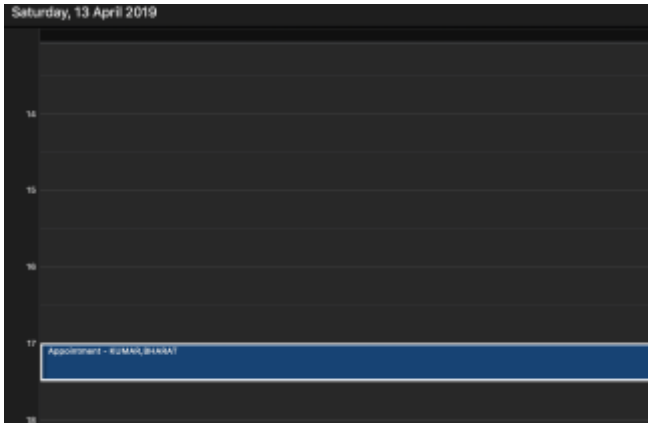

Fig.7 Calendar appointment created by Compute Engine

JSON (JavaScript Object Notation) is a lightweight data-interchange format. It is easy for humans to read and write. It is easy for machines to parse and generate. It is based on a subset of the JavaScript Programming Language, Standard ECMA-262 3rd Edition - December 1999. JSON is a text format that is completely language independent but uses conventions that are familiar to programmers of the C-family of languages, including C, C++, C#, Java, JavaScript, Perl, Python, and many others. These properties make JSON an ideal data-interchange language. [7]

JSON Objects are analyzed as string arrays, permitting higher parsing efficiency and easier preparation than heavier transport formats such as XML.

The response provided is maintained in a JSON Structure, as future enhancements could lead to this JSON being returned as a response to specific API requests externally. This response could be used to intelligently make pattern-based analysis and increase the overall scope of the service. However, it should be noted that this JSON structure doesn't provide any form of security against sensitive data directly. So, the JSON response needs to be encrypted when being routed to any service outside the environment that hosts the Digital Assistant.

This encryption can be achieved using JSON Web Encryption (JWE). JSON Web Encryption (JWE) is an IETF standard providing a standardized syntax for the exchange of encrypted data, based on JSON and Base64. It is defined by RFC7516. JWE forms part of the JavaScript Object Signing and Encryption (JOSE) suite of protocols [11].

VI. CHALLENGES IN PROPOSED APPROACH

Any software written is only good if it doesn't contain any obvious loop holes. Looking at certain challenges faced while implementing this approach:

 DOI : 10.35291/2454-9150.2019.0041

- The speech-to-text services offered are not always very quick to pick up conversations.
- Clear articulation of words is very important for relevance in creating proper data.
- Insufficient test data can result in skewed training routines for the learning engine.
- The speech-to-text engine struggles to pick up different accents and, misconstrues it as invalid words.
- The JSON response provided by the Learning Engine is not fully utilized, as many metrics remain unproductive in the current scenario.
- Many defining characteristics of human intelligence – pertaining to machine learning, which developed under much different pressures, remain out of reach for current approaches, such as generalizing beyond one's experiences.

VII. CONCLUSION AND FUTURE SCOPE

The digital assistant service can possibly be provided through many infrastructure platforms as an individual micro-service too.

The idea of creating an artificial intelligence powered digital assistant seems very simple but, one that involves many intricacies. The selected architecture format was not fully supportive of the actions needed to be performed. However, after some work arounds, the goal could be achieved.

The Digital assistant proves to require many iterations of training to actually be of good help. A future scope would be to feed it many iterations of training audio data so that it can perform efficiently as soon as it is used productively as a service.

Another future scope would be to provide a dashboard for a user to validate all the events picked up by the service, and to accept, edit, or delete the calendar events, as per the necessity.

REFERENCES

[1] Imrie, Peter & Bednar, Peter. (2013). Virtual Personal Assistant

[2] Voipstudio

[3] S Milhorat, Pierrick & Schlögl, Stephan & Chollet, Gerard & , Boudy & Esposito, Anna & Pelosi, G. (2014). Building the Next Generation of Personal Digital Assistants. 2014 1st International Conference on Advanced Technologies for Signal and Image Processing, ATSIP 2014. 10.1109/ATSIP.2014.6834655.

[4] Geeksforgeeks for Service oriented Architecture (SOA)

[5] Mathworks for Machine Learning

[6] Louisdorard, BigML and Data Science concepts

[7] Introducing JSON, Head First Series, 1st Edition

[8] Relational inductive biases, deep learning, and graph networks - Peter W. Battaglia

[9] Artificial Intelligence, McGraw Hill Education; 3 edition, Rich and Kevin.

[10] VoIP Handbook: Applications, Technologies, Reliability, and Security, Syed and Ilyas, CRC Press.

[11] JSON Web Encryption, Web Security, Wikipedia.

[12] Software Engineering, Pearson Publication, Ian Sommerville

[13] Digital Assistants, TESOL Strategy Guide Book by David Kent, Pedagogy Press

[14] Artificial Intelligence: A Modern Approach, 3e by Russel, Pearson Publication

[15] Artificial Intelligence Simplified: Understanding Basic Concepts by Binto George, Gail Carmichael, et al. | 11 January 2016

AUTHOR'S PROFILE

Mr. Raghav Sridhar, has 3+ years of IT Industry experience and currently employed as a back-end developer at SAP Labs India Pvt. Ltd. His areas of specialization are on backend technologies such as Spring framework, Java, Node JS, and certain front-end topics such as Javascript, SAP UI5.

Mr. Parthasarathy PD, has 3+ years of IT Industry experience and currently employed as a full-stack developer at SAP Labs India Pvt. Ltd. His areas of specialization are on front-end technologies such as MEAN Stack, JavaScript, OOJS, SAP UI5 & backend topics such as SAP ABAP, C# and Java and is a Machine Learning Enthusiast. He has contributed to various open source projects and Innovation topics within SAP. He is also a passionate trainer and coaches' newbies on various technical topics of *Computer Science and technologies*. On an academic front, he is a gold medalist at graduation and post-graduation level and loves applied research and wishes to pursue a PhD soon.

Dr. Vinod Vijayakumaran has 15 years of IT industry experience and currently employed as Technical Project Manager with HANA Enterprise Cloud, Partner Centre of Expertise, at SAP Labs India Pvt. Ltd. He holds doctorate from Karunya University, Coimbatore in the field of Computer Applications. His research area is on processes and this thesis was on a combined software development model incorporating LEAN, Agile and Six Sigma methodologies. Vinod holds a patent from USPO on his tool 'Downtime Calculator'

 DOI : 10.35291/2454-9150.2019.0041