Smart Virtual Assistant,

Rajesh Hadiya[1], Abid Merchant[2], Rudra Makwana[3], Prof. Jalpa Ramavat[4]

[1,2,3]Student, in Computer Engineering Department at VGEC, Chandkheda Ahmedabad
[4]Asst. Prof. Jalpa Ramavat, Computer Engineering Department of VGEC, Chandkheda Ahemadabad.
Their email addresses are hadiarajesh007@gmail.com and jalpa.ramavat.2012@gmail.com

Abstract:Smart Virtual assistant is capable of understanding human language and then respond to it in different ways. Virtual Assistant will be able to do basic tasks like opening mobile applications, opening services like wifi/hotspot, making notes, setting reminders/alarms, navigation, scheduling of user's activities, notifying for various meeting/events etc. It is also capable of responding to user's questions which can be on any topic, these types of questions would be replied by Assistant by searching on the Internet for appropriate answer, for this Machine Learning algorithms will be used. So, our Virtual Assistant will be there for user in every situation. User can ask a query to assistant by speech/text which will be taken as an input string by the application. Response by the assistant can be in the form of some action, speech response, graphical representation of some result or in the form of text, this depends of the type of query user had spoken.

I. INTRODUCTION

Our project is about a Virtual Assistant who is smart enough to understand human's Natural Language. It's response can be divided into two parts according to types of queries such as:

i. Action
ii. Question

If a user's query demands for an action, then assistant will directly perform that task, for example if user speaks "Open Whatsapp", then assistant will open whatsapp. These types of queries and bounded queries mean they don't need any further interaction.

Second type of query includes a question, in this type user asks assistant a question and in return expects an answer. For example, user questions "Who is Prime Minister of India" then assistant will respond as "Narendra Modi", these types of questions will be unbounded so in this case user may again ask further question relevant to above question.

Apart from these, assistant will be able to perform tasks like Navigation, News Updates, Interaction, Suggestions related to user etc. Figure shows architecture for Natural Language [2][4].

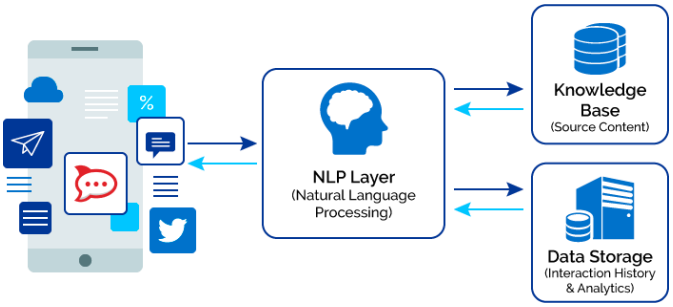

Figure-1 Natural Language Processing Architecture

II. PRINCIPLE OF OPERATION

Language Understanding

In the problem of Language Understanding is related to query understanding [1][3], subject to the constraints of the back-end data sources and the applications in terms of the filters they support and actions they execute. In practice, LU parses the string and analysis it to give results to user, results are not generated according to LU rules every time rather it gives results according to user experience. This is where the semantic schema comes into play, as it captures the constraints of the back-end knowledge sources and service APIs, while allowing free form of natural language expression to represent different user intents in an unambiguous manner. There are two main approaches to LU: rule based and machine learned. In rule-based approach, some rules are predefined by programmer and are updated according to requirement, this type of approach is static approach and the LU is not capable to change itself according to user experience. Machine Learned approach means that the assistant will learn from its experience and improvise itself rather than some fixed rules, it can be achieved by building a neural network and then training it by giving some sample tagged input. LU models are trained in a supervised fashion using labeled data and properly weighted lexicons. When a user issues a query, domain and intent classifiers are run to determine the domain and intent of the query, and the slot tagger tags semantic slots. Tagged slots are resolved into canonical values, and in some cases multiple slots are combined into a parameter. Parameters are used either to fetch results from the knowledge back end or to invoke an application API.

Knowledge Back End:

[1][3][6] One of the main parts of PDA is about accessing and fetching knowledge from backend. For example, when a user asks for facts regarding a movie or director, LU models tag such slots (i.e., facts) as movie name, release date, actors, and directors. Slots are used to build a query sent to these knowledge bases to fetch the relevant entity and relationships for which the user is looking. These knowledge bases store factual information in the form of entities and their relationships, covering many domains (people, places, sports, business, etc.). The entities and their relationships are organized using the World Wide Web Consortium Resource Description Framework (RDF). An RDF semantic knowledge base (also known as a *semantic knowledge graph*) represents information using triples of the form subject-predicate-object, where in graph form, the predicate is an edge linking an entity (the subject) to its attributes or another related entity. A popular open-source RDF semantic knowledge base is Freebase [5]. Other RDF semantic knowledge bases that are much larger in size are Facebook's Open Graph, Google's Knowledge Graph, and Microsoft's Satori. Both Google's Knowledge Graph and Microsoft's Satori knowledge graph have over 1 billion entities and many more entity relationships. They power entity-related results that are generated by Google's and Microsoft Bing's search engines. Recently, there has been a surge of interest in exploiting these knowledge sources, especially the RDF semantic knowledge bases, to reduce the manual work required for expanding conversational systems to generated by the domain, intent, and slot models to represent the complete semantic understanding of the query.

RNN

RNN is a kind of feed forward Neural Network [6] which contains a hidden layer/s and its activation is dependent on neurons of its preceding layer. Therefore, the RNN can handle variable-length sequences and model contextual information dynamically. Given an input sequence {x1, x2, ・ ・ ・ ,xT }, the standard RNN computes the output vector yt corresponding to each xt with the following equations:

ht = g(Wxhxt +Whhht−1 + bh)

yt = Whyht + by

where W denotes the weight matrix connecting two layers, in which Whh denotes the self-updating weights of the hidden layer. The term b denotes bias vector, while g(・) is an element-wise activation function such as sigmoid or relu functions depending on requirement, it is basically used as a squashing function to bound the output of neurons in a particular boundary. The graphical illustration of the standard RNN can be found in the left-hand side of Fig. 2. The bidirectional recurrent neural networks (BRNN) is proposed to access both the preceding and succeeding contexts by combining a forward hidden layer h and a backward hidden layer h as depicted in the right part of Fig. 2. The process can be expressed as follows:

$$\overrightarrow{h}_t = g(W_{x\overrightarrow{h}} x_t + W_{\overrightarrow{h}\overrightarrow{h}} \overrightarrow{h}_{t-1} + b_{\overrightarrow{h}})$$

$$\overleftarrow{h}_t = g(W_{x\overleftarrow{h}} x_t + W_{\overleftarrow{h}\overleftarrow{h}} \overleftarrow{h}_{t-1} + b_{\overleftarrow{h}})$$

$$y_t = W_{\overrightarrow{h}y} \overrightarrow{h}_t + W_{\overleftarrow{h}y} \overleftarrow{h}_t + b_y$$

The equations of both the RNN and the BRNN iterate from t=1 to T. The standard RNN is denoted as RNN in the sequel. Each hidden state can store all the input history information theoretically. However, they face the problem of "vanishing gradient" which states that the gradients tend to either vanish or explode exponentially, making the gradient based optimization method struggle. A lot of efforts are made to address the issue. LSTM is a popular solution for that.

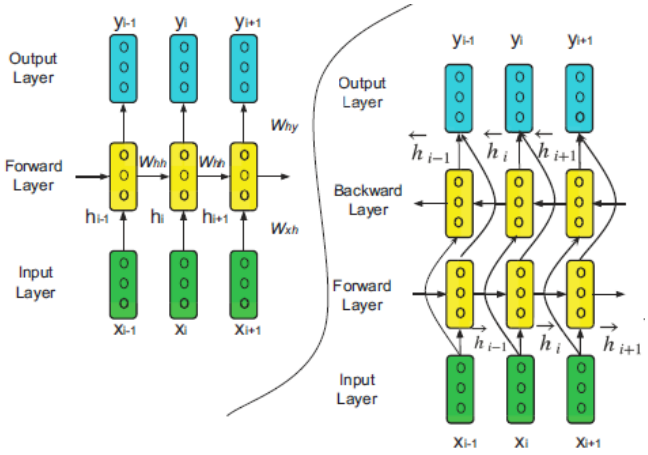

Figure-2 RNN working

LSTM

The LSTM addresses the problem of "vanishing gradient" by replacing the self-connected hidden units with memory blocks as illustrated. The memory units enable the network to be aware of when to learn new information and when to forget old information. We follow the implementation of the LSTM unit described [4]. The hidden state ht given input xt is computed as follows:

it = σ(Wxixt +Whiht−1 + bi)

ft = σ(Wxfxt +Whfht−1 + bf)

ot = σ(Wxoxt +Whoht−1 + bo)

ct = tanh(Wxcxt +Whcht−1 + bc)

ct = ft.ct−1 + it.ct

ht = ot.tanh(ct)

Where 'i' is an input gate modulating how much new memory content is added to the memory, 'f' is a forget gate modulating how much existing memory is forgotten, and 'o' is an output gate modulating the amount of memory content exposure. The memory cell ct consists of two components, namely partially forgotten previous memory ct−1 and modulated new memory ct. σ(・) and tanh (・) are sigmoid and hyperbolic tangent function respectively. As σ(x) = (1+e−x)−1 and tanh(x) = e^x−e−xe^x+e−x , all the gate values and hidden layer outputs lie within the range of [0, 1]. The operator denotes element-wise multiplication. The LSTM is able to handle more complex and longer temporal dynamics than the RNN by introducing these gates.

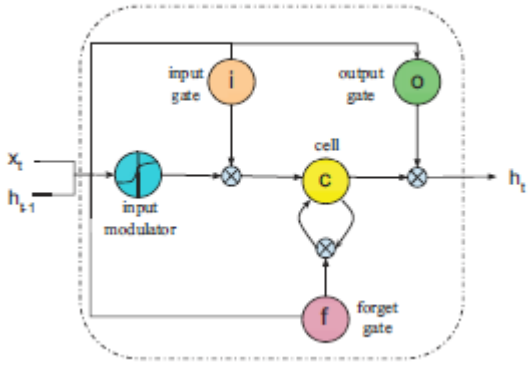

Figure-3 LSTM architecture

III. PERFORMANCE CHARATERISTIC

Basically performance of Assistant system depends upon following properties:

Speech Recognizer: It is the software code that capture the voice signals and convert it into text to make it understandable for machine.

Rules: It is regular expression rules which is used to parse the given data and analyze it to produce output.

Neural Network Architecture: It is composed of input, hidden and output layers to understand meaning of statement and produce output. Output depends upon use of Activation function like Sigmoid or ReLU.

SYSTEM DESIGN

The major technologies we are going to use are:

Android Studio: It is integrated development environment for Android application development. It provides code analysis, graphical debugger, integrated unit tester,
A Real-like android Emulator for app testing.

Tensorflow: It is an open-source software library for machine learning across a range of tasks. It is a system for building and training neural networks to detect and decipher patterns and correlations. Tensorflow-lite is mobile-optimized version of tensorflow.

Technology used by this system are mentioned below:
Natural Language Processing: We'll use Recurrent Neural Network (RNN) with Long-Short Term Memory (LSTM) for processing natural language.

Regular Expressions: We'll use regular expression to parse the basic actions on device.
Machine learning: It is application of artificial Intelligence and it automatically learn without programmed explicitly.

Algorithm:

The semantic frame for the query: "remind me to call my momat 9 a.m."[1]

```
<Domain score="0.8956">reminder</Domain>
<Intent score="0.7949">create_single_reminder</intent>
<Slots>
<reminder_text="call my mom" />
<start_time<RawValue="9 am">
<PropertyGroup Name="timex3">
<Property Name="value" Value="am" />
<Property Name="comment" Value="am" />
</PropertyGroup>
</start_time>
</Slots>
```

Hardware-Software used

Python I.D.E.
NLP tool kit
Android Studio
Android Device
Tensorflow

IV. ANALYSIS

In today's world, most of assistant system available in market is based on client-server architecture which means it requires an active internet connection to understand speech and analyze it to produce output. Further, most system only understand if predefine set of words is spoken.
Based on our analysis, we analyze that we have to make assistant which mostly operate on user-device until some internet related queries is given. and it should also be able to understand the user's speech without specific set of words.

V. APPLICATIONS

Our main objective of this project is to assist user in its day-to-day activities which in return will increase his/her productivity. Moreover, basic tasks will be automated and user can also get information from Internet just by asking it to Assistant.

VI. CONCLUSION

Currently, we are able to activate assistant by hot word (i.e. from any screen) and analyze simple speech and perform

actions according to them on device. But it is difficult to analyze complex query without use of machine learning techniques, so we are moving towards implementation of Recurrent neural network which can better understand and process speech and produce desired output.

VII. REFERENCE

[1] http://ieeexplore.ieee.org/document/7814394/

[2] https://www.xenonstack.com/blog/data-science/overview-of-artificial-intelligence-and-role-of-natural-language-processing-in-big-datahttps://pages.mtu.edu/~suits/notefreqs.html

[3] http://ieeexplore.ieee.org/document/8286763/

[4] https://patents.google.com/patent/US8346563B1

[5] K. Bollacker, C. Evans, P. Paritosh, T. Sturge, and J. Taylor, “Freebase: A collaborativelycreated graph database for structuring human knowledge,” in Proc. 2008 ACM SIGMOD Int. Conf. Management of Data, New York, 2008, pp. 1247–1250.

[6] http://ieeexplore.ieee.org/document/7727384/