ISSN 2395-1621

Future Generation of Personal Digital Assistants (Specifically for Android Based Smart Devices)

[#1]Hardik Pandya, [#2]Pravin Padole, [#3]Prafulla Patil, [#4]Amruta Pawar

[1]pandyanhardik@gmail.com
[2]pravinpadole7@gmail.com
[3]prafull.patil310@gmail.com
[4]amrutapawar15j@gmail.com

[#1234]Student, Computer Engineering

TSSM's Bhivarabai sawant Collage of Engg & Research
Narhe, Pune, India.

ABSTRACT

Voice Command based Personal Digital Assistants are Getting really popular these days like Siri & Google Now for smart phones. They usually promise to provide context aware solution for user's distinct needs. But Heay. Real Assistants are still missing! In this paper we're going to describe some of challenges building those personalized voice operated assistive Apps. Many research and ways we have learned in order to solve the problem. In fact we'll primarily focus on Natural Language Processing (NLP) & Dialogue Management as we believe, these side of technology requires more attention to actually work on.

Keywords: **Personal Digital Assistants; Natural Language Processing; Speech Recognition; Natural Dialog Understanding; Intent Management.**

ARTICLE INFO

Article History

Received: 27th May 2016

Received in revised form :

27th May 2016

Accepted: 30th May 2016

Published online :

31st May 2016

I. INTRODUCTION

Recent trend indicates slightly increasing the development in speech enabled Personal Digital Assistants. During past 20 years, the necessary technology inventions (i.e. Composition of Natural Language Processing, Auto Speech synthesis, Dialog management, Text-speech Processing) have been mainly focused on academic and industrial result, resulting in commercial products such as Apple's siri and Google's Now & api.ai's Speak to It Assistant. These developments are not only the avidence of advancement in technology innovations but also manages to bring the concept of Artificially Intelligent personal assistants i.e. a system that is able to (at some extend) understand and respond to spoken voice commands, somewhat at end user side. Rapid improvement happened from tech perspective as well as to supported application domains.

The very first system that used natural language processing were mainly focused on travel domain, The possible application scenario have progressively been changed and now involves domains such as weather forecast, navigation, translation, information, tutoring, HealthCare etc. Despite to the overall enhancement in the use of natural language driven assistants is however still continues. May be, only one reason for resistance found in the reality that Siri Inc. all their hard work is not really for personal use. Main focus on personal adaptation is necessary to better integrate systems with users' daily tasks. While the technology has been already reached the point where it may be called robust, but the integration of context and personalized behavior is still at its Initial stage. Significant improvements are still needed in order to move from pure PDAs to true personal Assistants.

II. PERSONAL DIGITAL ASSISTANTS

Our common understanding of a personal assistant is that of a person (or an agent) who is able to provide distinct help at a given time and in a given activity context. For example, a secretary situated in a general work context provides support for activities such as answering incoming calls, recording meetings and appointments, ordering products, or interacting with clients. An important characteristic of personal assistants is that they adapt to the distinct demands of their 'master' and furthermore (over time) progressively pay attention to her/his personal preferences and routines. Also, as by their definition, personal assistants should each be helping only one person, making this one-to-one relationship between assistant and 'master' a crucial benefit. Given this definition of a personal assistant it is easy to argue that many people would find it convenient to have such a person at their

© 2015, IERJ All Rights Reserved

disposal, even though not all of them may have a clear understanding of their exact context of use. Hence, when we try to approach this subject from a more technical perspective, we already find a set of requirements and expectations coming from users. Among the most requested features when thinking of digital assistants, are simplicity, flexibility and easiness of interaction. Voice-based input/output interfaces may be the easiest way to fulfill these requirements. This is because 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.

III. POTENTIAL AREAS OF IMPROVEMENT

Given the system requirements described above and we see four areas where significant improvements are possible:

A. Extended Dialog History

Deployed systems, whether they have a research or a commercial purpose, usually utilize the dialog history as an instrument to authorize user utterances and to keep track of the dialog state. the memorized activity log is often focused on a single dialog and does not spread across different conversations. Expanding the history element may allow for building systems that adapt to the user over time. Hence, systems should be able to adapt the speech segment's acoustic models, the language models, the understanding as well as the general way of completing a dialog. For example, the overall interaction could be improved if the system has information about the identity, the gender and the age of a user, and further more respects his/her preferences for interacting with technology.

B. Improved Context Awareness

In addition to what is directly requested by the user, a vast amount of information is usually available which may be processed by a system and eventually could improve its context awareness. For example, sensors embedded in (future) homes, in the office or simply in every day used appliances such as our beloved smartphones, are precious data sources that could be used to augment human-machine interaction.Also the Internet may be a potential data pool that could be mined in order to provide an enhanced knowledge of the "outside world" and consequently improve a system's reasoning.

C. Dynamic System Adaptation

The improvement and usage of a SDS for the most part comprises of three stages. the specification of the relevant components is The first stage is concerned . the implementation is the second and the actual deployment of the system is third. From the implementation stage onward, a system's configuration is often static, i.e. it does not change with usage nor according to a given dialog state. We therefore propose to use a multi-agent architecture so that the settings of one agent can be changed based on the input coming from other agents. So if one component detects a change in the dialog context it can inform all the other components, updating their configuration. This dynamic mechanism would allow for modifying the system even while it is running.

D. Supported Task Hierarchy Design

Previous work has highlighted an overall classification for Dialog Management (DM) paradigms. According to this classification a DM is based on either stochastic processes i.e. MDPs, POMDPs, the information state principles, or a hierarchy of tasks . Each of these categories has advantages and drawbacks. The information state paradigm is considered as an inaccurate theoretical framework and therefore does not meet the requirements of most practical implementations. The same applies to stochastic processes whose main drawback is that training data has to be collected in order to build a system. Despite significant research efforts to overcome this issue (e.g. reinforcement learning) these processes are yet to be streamlined. The task-based paradigm therefore remains the most appropriate technique. Depending on the size of a dialog, the initiative given to the user, and the conditional execution steps, task hierarchies may often be very complex. Better tools and methods should therefore be available so that the design of task hierarchies and their automatic transformations can be shifted to the specification stage of an application.

IV. A PROPOSED ARCHITECTURE

In order to discussed areas of improvement, we hab been implemented a Spoken Dialog System (SDS). Its architecture are follow in Fig. 1. The illustrated base components represent specialized agents which communicate with each other using the Active MQ messaging server5. The data flow between components is restricted so as to control for a near-sequential processing pipeline. Currently this is achieved with a single sensor i.e. a microphone that captures the signal, which is subsequently analyzed by the ASR engine The ASR frames the signal, converts it to a sequence of finite observation vectors and matches those against a set of acoustic and linguistic stochastic models. It produces a ranked list of hypotheses for the text content recognized from the speech signal and associates them with confidence scores related to their acoustic likelihood and linguistic probability. Only transcriptions with scores passing a given threshold are propagated to the core system. In the case that no transcription passes the threshold a generic "not understood" message is passed on. On the other end of the pipeline the TTS engine generates and plays audio responses from text sent by intermediate processing component.

This system layer links the signal level with the textual level of the architecture. Additional sensors may be integrated in the future so as to increase the entropy of the captured information and consequently improve the system's overall performance. For the moment, however, the focus lies on improving the middle components dealing with language understanding, dialog management and text generation. Here the Natural Language Generator (NLG) is currently based on a fairly simple template selection process which randomly chooses text utterances according to the current semantic representation of the system. The Natural Language Understanding

(NLU) component, however, is a complex association of several integrated components. Section 5 will describe these components in more detail as well as the successive transformations that have to be applied in order to process the text transcriptions coming from the speech recognizer. Finally, the center piece of the proposed architecture is the dialog manager [14]. It is based on a generic inference engine that has to be configured using scenario-based task hierarchies. It executes actions and generates semantic concepts related to user inputs. Also here the reader will find

© 2015, IERJ All Rights Reserved

a more detailed description of the component, and the way it is set and used by the system, in Section 5.

V.COMPONENTS AND ALGORITHMS

In this section, we present the components and algorithms we propose in order to tackle the challenges described in Section 3 First, a discussion of the natural language understanding components is provided followed by a clarification of the types of problems we intend to solve with this setup. Then, the implementation of a dialog model design tool is detailed. Overall the natural language understanding process aims at providing a mapping between the unrestricted infinite word-level semantic space a user may utilize when interacting with a system, and the dynamic set of parameterized dialog acts a system is capable of dealing with at any given point in a dialog. As such natural language understanding is a multi-step dynamic process with clearly separated roles taken on by individual sub-components. The input space size of any sub-component is greater or equal to the size of its output space. In other words, an NLU engine processes user utterances sequentially and, at every stage, the level of the semantics extracted from the text is more machine oriented. This sequential processing of a transcription until it reaches the dialog manager is illustrated in Fig. 1.

A. Parsing

A semantic parser connected semantic labels with a text utterances. parsing techniques are based on context-free grammars. which are hand-coded based on the analysis of collected dialog data, and designed by experts.
For our system we integrated the algorithm. Instead of matching complete-sentence patterns with text input, proposeto look for patterns in chunks of the text-level sentence and in the temporary (i.e. already assigned) Semantic Frame (SF).
An operation is applied to the current SF that time the triggering pattern is detected. The rules consisting of a trigger part (the pattern to look for) and an operation part (the transformation to apply) are automatically learned. At every step of the training process, the algorithm tests all the pairs available.

Unification

As of now specified, the parsing in light of standards got from the earlier logged off preparing and a client's content input.that not handling any logical data. Along these lines parsing results might be shallow. Case in point, a client articulation, for example, "two" would be named as "input:number=2", where information is the objective of the semantic casing and "2" is the estimation of the space "number". The articulation itself does not shoulder more data, which demonstrates the requirement for different wellsprings of relevant data to increase the essential semantic representation. We have in this manner characterized a module for Semantic Unification and Reference Resolution (SURR). The SURR keeps up a dynamic tree gathering whose hubs are sets of openings (some of them esteemed). The edge between a guardian hub and one or more youngsters node(s) bears a molded change that applies to the worth or the name (or both) of the spaces in the kids hubs while going up the trees. All together for a semantic casing to breeze through the test of this sifting part, as of now specified, the parsing depends exclusively on principles got from the earlier logged off preparing and a client's content data, without handling any relevant data. Hence parsing results might be shallow.

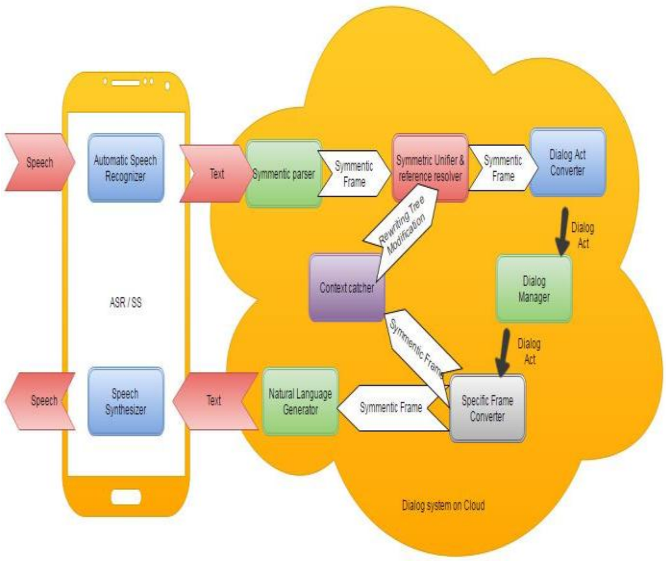

Fig 1. The Proposed System Architecture

VI. CONCLUSION AND FUTURE WORK

The proposed methodology examined in this paper go for making a compelled human-machine dialog more adaptable and versatile to the client's necessities, bypassing the restrictions of the current mechanical capacities. We talked about a few difficulties for building future SDS-driven Personal Advanced Assistants, and depicted various segments also; calculations with which we expect to handle them. Our objective for what's to come is to assist enhances our answers and makes them accessible, so that likewise different frameworks may actualize them. The test of meeting advanced certifiable dialog necessities and additionally the unpredictability of specific relevant occasions recommend that, for quite a while, it will be doubtful to build up any standard for planning powerful and viability dialog frameworks, even in natural compelled applications. The thoughts proposed in this paper ought to thusly be considered as an endeavour to advance towards better, more adaptable characteristic dialect client interfaces.

ACKNOWLEDGMENT

The authors would like to acknowledge the Prof. Sonali Pawar at TSSM's Bhivarabai Sawant Collage of Engineering And Research. for providing B.E Final Year project guidance in Building of Personal Digital Assistant for Android Platform and Opulent InfoTech For sponsoring our Project.

REFERENCES

[1] Punjabi, Relaxed Context-Aware Machine Learning Midddleware (RCAMM) for Android. IEEE Press Wiley: IEEE, 2013.

[2] H. Arsikere, Computationally-Efficient End pointing Features For Natural Spoken Interaction With Personal-Assistant Systems. IEEE Press Wiley: IEEE, 2014.

© 2015, IERJ All Rights Reserved

[3] A. Mishra, A Voice-Controlled Personal Assistant Robot. IEEE Press Wiley: IEEE, 2015.

[4] P.Milhorat, Building The Next Generation Of Personal Digital Assistants. IEEE Press Wiley: IEEE, 2014.

[5] J. Jiang, “How do users respond to voice input errors? lexical and phonetic query reformulation in voice search,” 2013.

[6] D.Javale, Home automation and security system using Android ADK. IJECCT.

[7] S.Kumar, Ubiquitous smart home system using android application. Wiley: IJCNC, 2014\

[8] V. Zue, S. Seneff, J. R. Glass, J. Polifroni, C. Pao, T. J. Hazen, and L. Hetherington, “J UPITER : A Telephone-Based Conversational Interface for Weather Information,” IEEE Transactions on Speech and Audio Processing, vol. 8, no. 1, pp. 85–96, 2000.

[9] R. Belvin, R. Burns, and C. Hein, “Development of the HRL route navigation dialogue system,” in Proceedings of ACL-HLT, 2001

[10] M. Kolss, D. Bernreuther, M. Paulik, S. St¨ucker, S. Vogel, and A.Waibel, “Open Domain Speech Recognition & Translation: Lectures and Speeches,” in Proceedings of ICASSP, 2006.

[11] S. M¨oller, J. Krebber, A. Raake, P. Smeele, M. Rajman, M. Melichar, V. Pallotta, G. Tsakou, B. Kladis, A. Vovos, J. Hoonhout, D. Schuchardt, N. Fakotakis, T. Ganchev, and I. Potamitis, “INSPIRE: Evaluation of a Smart-Home System for Infotainment Management and Device Control,” in Proceedings of LREC, 2004, pp. 1603–1606.

[12] I. Braun and N. Rummel, “Facilitating Learning From Computer-Supported Collaborative Inquiry: the Challenge of Directing Learners’ Interactions To Useful Ends,” Research and Practice in Technology Enhanced Learning, vol. 05, no. 03, pp. 205, 2010.

© 2015, IERJ All Rights Reserved