VOICE BASED SELF HELP SYSTEM: USER EXPERIENCE VS ACCURACY

Sunil Kumar Kopparapu

TCS Innovation Lab - Mumbai
Tata Consultancy Services, Yantra Park, Thane (West), Maharastra, India.
Email: SunilKumar.Kopparapu@TCS.Com

ABSTRACT

In general, self help systems are being increasingly deployed by *service based* industries because they are capable of delivering better customer service and increasingly the switch is to voice based self help systems because they provide a *natural* interface for a human to interact with a machine. A speech based self help system ideally needs a speech recognition engine to convert spoken speech to text and in addition a language processing engine to take care of any misrecognitions by the speech recognition engine. Any off-the-shelf speech recognition engine is generally a combination of acoustic processing and speech grammar. While this is the norm, we believe that ideally a speech recognition application should have in addition to a speech recognition engine a *separate* language processing engine to give the system better performance. In this paper, we discuss ways in which the speech recognition engine and the language processing engine can be combined to give a better user experience.

Index Terms— User Interface; Self help solution; Spoken Language System, Speech Recognition, Natural Language

1. INTRODUCTION

Self help solutions are being increasingly deployed by many service oriented industries essentially to serve their customer-base any time, any where. Technology based on artificial intelligence are being used to develop self help solutions. Typically self help solutions are web based and voice based. Oflate use of voice based self help solutions are gaining popularity because of the ease with which they can be used fueled by the significant development in the area of speech technology. Speech recognition engines are being increasingly used in several applications with varying degree of success. The reason businesses are investing in speech are several. Significant reasons among them are the return on investment (RoI) which for speech recognition solutions is typically $9 - 12$ months. In many cases it is as less as 3 months [1]. In addition, speech solutions are economical and effective in improving customer satisfaction, operations and workforce productivity. Password resetting using speech [2], airline enquiry, talking yellow pages [3] and more recently in the area of contact centers are some of the areas where speech solutions have been demonstrated and used.

The increased performance of the speech solution can be primarily attributed to several factors. For example, the work in the area of dialog design, language processing have contributed to the performance enhancement of the speech solution making them deployable in addition to the fact that people have become more comfortable using voice as an interface to transact now. The performance of a speech engine is primarily based on two aspects, namely, (a) the acoustic performance and (b) the non-acoustic performance. While the change in the acoustic performance of the speech engine has increased moderately the mature use of non-acoustic aspects have made the speech engine usable in applications; a combination of this in total enables good user experience.

Any speech based solution requires a spoken speech signal to be converted into text and this text is further processed to derive some form of information from an electronic database (in most practical systems). The process of converting the spoken speech into text is broadly the speech recognition engine domain while the later, converting the text into a meaningful text string to enable a machine to process it is the domain of natural language (NL) processing. In literature, these two have a very thin line demarcating them and is usually fuzzy, because the language processing is also done in the form of speech grammar in a speech recognition engine environment. This has been noted by Pieraccini et al [4], where they talk about the complexity involved in integration language constraints into a large vocabulary speech recognition system. They propose to have a limited language capability in the speech recognizer and transfer the complete language capability to a post processing unit. Young et al [5] speak of a system which combines natural language processing with speech understanding in the context of a problem solving dialog while Dirk et al [6] suggest the use language model to integrate speech recognition with semantic analysis. The MIT Voyager speech understanding system [7] interacts with the user through spoken dialog and the authors describe their attempts at the integration between the speech recognition and natural language components. [8] talks of combining statistical and knowledge based spoken language to enhance speech based solution.

In this paper, we describe a non-dialog based self help system, meaning, a query by the user is responded by a single answer by the system; there is no interactive session between the machine and the user (as in a dialog based system). The idea being that the user queries and the system responds with an answer assuming that the query is complete in the sense that an answer is *fetchable*. In the event the query is incomplete the natural language processing (NL) engine responds with a close answer by making a few assumptions, when required [9]. The NL engine in addition also *corrects* any possible speech engine mis-recognition. In this paper, we make no effort to distinguish the differences in the way language is processed in the speech recognition module and the natural language processing module. We argue that it is optimal (in the sense of performance of the speech solution plus user experience) to use as a combination of language model in the speech recognition and natural language processing modules. We first show (Section 2) that language processing has to be distributed and can not be limited to either the speech recognition or the natural language processing engine. We then show how using a readily available SR engine and a NL engine [9] how the distribution of language processing helps in proving better user experience. We describe a speech based self self help system in Section 3 and describe the user experiences in Section 4. We conclude in Section 5.

2. BACKGROUND

For any speech based solution to work in field there are two important parameters, namely, the accuracy of the speech recognition engine and the overall user experience. While both of these are not entirely independent it is useful to consider them as being independent to be able to understand the performance of the speech solution and the user experience associated with the solution. User experience is measured by the freedom the system gives the user in terms of (a) who can speak (speaker independent), (b) what can be spoken (large vocabulary) and (c) how to speak (restricted or free speech) while the speech recognition accuracies are measured as the ability of the speech engine to convert the spoken speech into *exact* text.

Let $\mathbb{F}$ represent speech recognition engine and let $\mathbb{E}$ be the natural language processing engine. Observe that $\mathbb{F}$ converts the acoustic signal or a time sequence into a string sequence (string of words)

$$\mathbb{F} : \text{time sequence} \rightarrow \text{string sequence}$$

while $\mathbb{E}$ processes a string sequence to generate another string sequence.

$$\mathbb{E} : \text{string sequence} \rightarrow \text{string sequence}$$

Let q_t represents the spoken query corresponding to, say, the string query q_s (it can be considered the read version of the written string of words q_s). Then the operations of the speech and the natural language processing engines can be represented as

$$\begin{aligned} \mathbb{F}(q_t) &= q_{s'} \text{ (speech engine)} \\ \mathbb{E}(q_{s'}) &= q_{s''} \text{ (NL processing)} \end{aligned} \quad (1)$$

Clearly, the speech recognition engine $\mathbb{F}$ uses acoustic models (usually hidden Markov Model based) and language grammar which are tightly coupled to convert q_t to $q_{s'}$ while the natural language engine $\mathbb{E}$ operates on $q_{s'}$ and uses *only* statistical or knowledge based language grammar to convert it into $q_{s''}$. It is clear that the language processing happens both in $\mathbb{F}$ and $\mathbb{E}$ the only difference being that the language processing in $\mathbb{F}$ is tightly coupled with the overall functioning of the speech recognition engine unlike in $\mathbb{E}$. Language processing or grammar used in $\mathbb{F}$ is tightly coupled with the acoustic models and hence the degree of configurability is very limited (speech to text). At the same time language processing is necessary to perform *reasonable* recognition (speech recognition performance). While there is a relatively high degree of configurability possible in $\mathbb{E} : q_s \rightarrow q_s$ (text to text). The idea of any speech based solution is to build $\mathbb{F}$ and $\mathbb{E}$ such that their combined effort, namely, $\mathbb{E}(\mathbb{F}(q_t)) = q_{s''}$ is such that $q_{s''} \approx q_s$. Do we need language processing in both $\mathbb{F}$ and $\mathbb{E}$ or is it sufficient to (a) isolate $\mathbb{F}$ and $\mathbb{E}$; and have language processing only in $\mathbb{E}$ or (b) combine all language processing into $\mathbb{F}$ and do away with $\mathbb{E}$ completely. Probably there is an optimal combination of $\mathbb{F}$ and $\mathbb{E}$ which produces a usable speech based solution.

An ideal speech recognition system should be able to convert q_t into the exact query string q_s. Assume that there are three different types of speech recognition engines. Let the speech recognition engine $\mathbb{F}_1$ allow any user to speak anything (speaker independent dictation system); $\mathbb{F}_2$ be such that it is $\mathbb{F}_1$ but the performance is tuned to a particular person (person dependent) and $\mathbb{F}_3$ is such that it is $\mathbb{F}_2$ additionally constrained in the sense that it allows the user to speak from within a restricted grammar. Clearly the user experience is best for $\mathbb{F}_1(x_t) = x^1_{s'}$ (user experience: $\uparrow$) and worst for $\mathbb{F}_3(x_t) = x^3_{s'}$ (user experience: $\downarrow$) and it between experience is provided by $\mathbb{F}_1(x_t) = x^2_{s'}$ (user experience: $\leftrightarrow$).

Let $d(x_s, y_s)$ be the distance between the string x_s and y_s. Clearly, $d(x^1_{s'}, x_s) > d(x^2_{s'}, x_s) > d(x^3_{s'}, x_s)$, the performance of the speech engine is best for $\mathbb{F}_3$ followed by $\mathbb{F}_2$ followed by $\mathbb{F}_1$. Observe that in terms of user experience it is the reverse. For the overall speech system to perform *well* the contribution of $\mathbb{E}$ would vary, namely $\mathbb{E}$ should be able to generate $q^1_{s''}$, $q^2_{s''}$ and $q^3_{s''}$ using $q^1_{s'}$, $q^2_{s'}$ and $q^3_{s'}$ respectively, so that $d(q^1_{s''}, q_s) \approx d(q^2_{s''}, q_s) \approx d(q^3_{s''}, q_s) \approx 0$. The performance of $\mathbb{E}$ has to be better to compensate for the *poor* performance of $\mathbb{F}$; for example the performance of $\mathbb{E}_1$ has to be better than the performance of $\mathbb{E}_3$ to compensate for the poor performance of $\mathbb{F}_1$ compared to $\mathbb{F}_3$.

Typically, a $I\!F_1$ (ideal user experience) speech recognition would be categorized by (a) Open Speech (free speech - speak without constraints), (b) Speaker independent (different accents, dialects, age, gender) and (c) Environment independent (office, public telephone). While (a) greatly depends on the language model used in the speech recognition system, both (b) and (c) depend on the acoustic models in the SR. For $I\!F_1$ type of system, the user experience is good but speech recognition engine accuracies are poor. On the other hand, a typical $I\!F_3$ (bad on user experience) would be categorized by limiting the domain of operation and the system would be tuned (in other words constrained) to make use of prior information on expected type of queries.

In the next section we describe a voice based self help system which enables us to tune the language grammar and hence control the performance of the speech recognition engine.

3. VOICE BASED SELF HELP SYSTEM

Voice based self help system is a speech enabled solution which enables human users to interact with a machine using their speech to carry out a transaction. To better understand the role of $I\!F$ and $I\!E$ in a speech solution we actually built a voice based self help system. The self help system was built using the Speech Recognition ($I\!F$) engine of Microsoft using Microsoft SAPI SDK [10] and the Language Processing ($I\!E$) module was developed in-house [9].

In general, insurance agents act as intermediaries between the insurance company (service providing company) and their clients (actual insurance seekers). Usually, the insurance agents keep track of information of their clients (policy status, maturity status, change of address request among other things) by being in touch with the insurance company. In the absence of a self help system, the insurance agents got information by speaking to live agents at a call center run by the insurance company. The reason for building a self help system was to enable the insurance company to lower the use of call center usage and additionally providing dynamic information needed by agents; both this together provide better customer service. The automated self help system, enabled answering queries of an insurance agent. Figure 1 shows a high level functional block representation of the self help system. The user (represented as a mobile phone in Figure 1) calls a predetermined number and speaks his queries to get information. The speech recognition engine converts the spoken query (speech signal) into text; this text is operated upon by the natural language processing block. This processed string is then used to fetch an answer from the database. The response to this query, which is a text string, is (a) spoken out to the user using a text to speech engine and (b) alternatively is sent to the user as a SMS. We used Kannel, an open source WAP and SMS gateway to send the answer string as SMS [11].

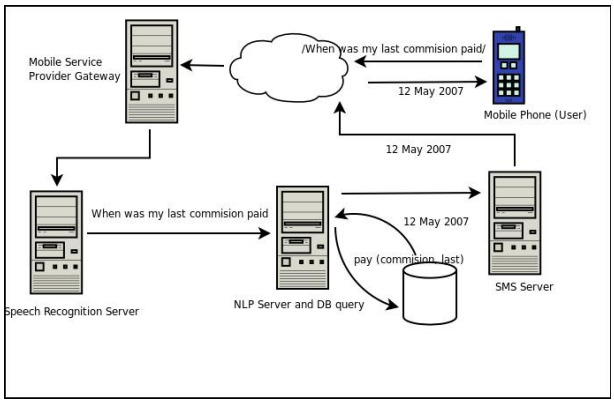

Fig. 1. Block Diagram of a Self Help System

The self help system was designed to cater to

1. different kinds of information sought by insurance agent
   (a) on behalf of their clients (example, *What is the maturity value of the policy TRS1027465*) and
   (b) themselves (example, *When was my last commission paid?*).
2. different accents,
3. handle different complexity of queries and
4. Additionally the system should be able to accept natural English query.
5. different ways in which same queries can be asked, Examples:
   (a) Surrender value of policy TRS1027465?
   (b) What is the surrender value of policy TRS1027465?
   (c) Can you tell me surrender value of policy TRS1027465?
   (d) Please let me know the surrender value of policy TRS1027465?
   (e) Please tell me surrender value of policy TRS1027465?
   (f) Tell me surrender value of policy TRS1027465?
   (g) My policy is TRS1027465. What is its surrender value?

should all be understood as being queried for the

... surrender_value of the *policy TRS1027465....*

Note that the performance of the speech recognition engine is controlled by the speech grammar (see Figures 2, 3, 4 for examples of speech grammar) that drive the speech recognition engine. The speech grammar is used by the speech engine before converting the spoken acoustic signal into a text string. In Section 4 we show how the performance of the speech engine can be controlled by varying the speech grammar.

4. EXPERIMENTAL RESULTS

We built three versions of the self help system with varying degrees of processing distributed in $I\!F$ and $I\!E$, namely,

1. $I\!F_1$ has no grammar (Figure 2), giving a very high degree of freedom to the user as to what they can ask, giving them scope to ask invalid queries.

2. $I\!F_2$ (Figure 3) has liberal grammar; more processing in $I\!E$ and

3. $I\!F_3$ has a constrained grammar (see Figure 4) which constraints the flexibility of what the user can say

For example, $I\!F_1$ grammar would validate even an out of domain query like *What does this system do?* in one dimension and an incorrect query like *What is last paid commission address change?*. On the other extreme a $I\!F_3$ grammar would only recognize queries like *What is the surrender value of Policy number* or *Can you please tell me the maturity value of Policy number* and so on. Note that the constrained grammar $I\!F_3$ gives a very accurate speech recognition because the speaker speaks what the speech engine expects this in turn puts very less load in terms of processing on $I\!E$.

For the experimental setup, the $I\!F_1$ grammar generated a total of 27 possible queries of which only 3 were not responded by the $I\!F_1$, $I\!E$ combined system. On the other hand for a grammar of type $I\!F_3$ a total of 357 different queries that the user could ask possible (very high degree of flexibility to the user). Of these only a total of 212 queries were valid in the sense that they were meaningful and could be answered by the $I\!F_3$, $I\!E$ system the rest, 145, were processed by $I\!E$ but were not meaningful and hence an answer was not provided. The performance of $I\!F_2$ grammar was in between these two cases producing a total of 76 possible queries that the user could ask, of which 20 were invalided by the $I\!F_2$, $I\!E$ combine.

```
<GRAMMAR>
 <RULE NAME="F_1" TOPLEVEL="ACTIVE">
  <RULEREF NAME="DonotCare"/>
 </RULE>
</GRAMMAR>
```

Fig. 2. $I\!F_1$: No grammar; the speaker can speak anything.

5. CONCLUSIONS

The performance of a voice based self help solution has two components; user experience and the performance of the speech engine in converting the spoken speech into text. It was shown that $I\!F$ and $I\!E$ can be used jointly to come up with types of self help solutions which have varying effect on the user experience and performance of the speech engine. Further, we showed that on one hand by controlling the language grammar one could provide better user experience but the performance of the speech recognition became poor while on the other hand when the grammar was such that the performance of speech engine was good the user experience became poor. This shows that there is a balance between the speech recognition accuracy and user experience that is to be maintained by people who design voiced based self help systems so that both the speech recognition accuracy is good without sacrificing the user experience.

```
<GRAMMAR>
 <RULE NAME="F_2" TOPLEVEL="ACTIVE">
  <RULEREF NAME="DonotCare"/>
  <RULEREF NAME="KeyConcept"/>
  <RULEREF NAME="DonotCare"/>
  <RULEREF NAME="KeyWord"/>
  <RULEREF NAME="DonotCare"/>
 </RULE>
 <RULE NAME="KeyConcept">
  <P> Surrender Value </P>
  <P> Maturity Value </P>
  <P> ... </P>
  <P> Address Change </P>
 </RULE>
 <RULE NAME="KeyWord">
  <P> Policy Number </P>
  <P> ... </P>
  <P> ... </P>
 </RULE>
</GRAMMAR>
```

Fig. 3. $I\!F_2$: Liberal grammar: Some restriction on the user.

6. REFERENCES

[1] Daniel Hong, "An introductory guide to speech recognition solutions," Industry white paper by Datamonitor, 2006.

[2] Microsoft Research, "Microsoft speech - solutions: Password reset," http://www.microsoft.com/ speech/ solutions/ pword/ default.mspx, 2007.

[3] Tellme, "Every day info," http://www.tellme.com/ products/ TellmeByVoice, 2007.

[4] Roberto Pieraccini and Chin-Hui Lee, "Factorization of language constraints in speech recognition," in *Proceedings of the 29th annual meeting on Association for Computational Linguistics*, Morristown, NJ, USA, 1991, pp. 299–306, Association for Computational Linguistics.

[5] S. L. Young, A. G. Hauptmann, W. H. Ward, E. T. Smith, and P. Werner, "High level knowledge sources in usable

```
<GRAMMAR>
 <RULE NAME="F_3" TOPLEVEL="ACTIVE">
  <o> <RULEREF NAME="StartTag"/> </o>
  <RULEREF NAME="KeyConcept"/>
  <o> of <o> the </o> </o>
  <o> in <o> the </o> </o>
  <RULEREF NAME="KeyWord"/>
  <o> <RULEREF NAME="EndTag"/> </o>
 </RULE>
 <RULE NAME="StartTag">
  <P> What is the </P>
  <P> Please send me </P>
  <P> Can you please send me</P>
  <P> Can you tell me </P>
 </RULE>
 <RULE NAME="KeyConcept">
  <P> Surrender Value </P>
  <P> Maturity Value </P>
  <P> ... </P>
  <P> Address Change </P>
 </RULE>
 <RULE NAME="KeyWord">
  <P> Policy Number </P>
  <P> ... </P>
  <P> ... </P>
 </RULE>
 <RULE NAME="EndTag">
  <P> Thank You </P>
  <P> ... </P>
 </RULE>
</GRAMMAR>
```

Fig. 4. $\mathbb{F}_3$: Constrained grammar - speaker is highly constrained in what he can speak.

speech recognition systems," *Commun. ACM*, vol. 32, no. 2, pp. 183–194, 1989.

[6] Dirk Buhler, Wolfgang Minker, and Artha Elciyanti, "Using language modelling to integrate speech recognition with a flat semantic analysis," in *6th SIGdial Workshop on Discourse and Dialogue*, Lisbon, Portugal, September 2005.

[7] Victor W. Zue, James Glass, David Goodine, Hong Leung, Michael Phillips, Joseph Polifroni, and Stephanie Seneff, "Integration of Speech Recognition and Natural Language Processing in the MIT Voyager System," in *Proc. ICASSP*, 1991, vol. 1, pp. 713–716.

[8] Ye-Yi Wang, Alex Acero, Milind Mahajan, and John Lee, "Combining statistical and knowledge-based spoken language understanding in conditional models," in *Proceedings of the COLING/ACL on Main conference poster sessions*, Morristown, NJ, USA, 2006, pp. 882–889, Association for Computational Linguistics.

[9] Sunil Kumar Kopparapu, Akhlesh Srivastava, and P. V. S. Rao, "Minimal parsing key concept based question answering system," in *HCI (3)*, Julie A. Jacko, Ed. 2007, vol. 4552 of *Lecture Notes in Computer Science*, pp. 104–113, Springer.

[10] Microsoft, "Microsoft Speech API," *http://msdn.microsoft.com/en-us/library/ms723627(VS.85).aspx*, Accessed Nov 2008.

[11] Open Source, "Kannel: Open source WAP and SMS gateway," *http://www.kannel.org/*, Accessed Nov 2008.