Today's Challenges for Embedded ASR

Jozef Ivanecký(✉) and Stephan Mehlhase

European Media Laboratory, Schloss–Wolfsbrunnenweg 35,
69118 Heidelberg, Germany
{jozef.ivanecky,stephan.mehlhase}@eml.org

Abstract. Automatic Speech Recognition (ASR) is pervading nowadays to areas unimaginable a few years ago. This progress was achieved mainly due to massive changes in the "Smart phones world" and ubiquitous availability of small, and powerful Linux-based hardware.

Recently, the availability of free ASR systems with acceptable speed and accuracy performance grew. Together with the changes the mobile world brought, a developer is now able to include ASR quickly without detailed knowledge of the underlying technology. What will be the future of embedded ASR systems in this case?

This talk presents two embedded ASR applications and points out their advantages over today's quick solutions. The first one demonstrates how changes in users behavior allowed to design a usable voice enabled house control. The second one is an extremely reliable in–car real–time ASR system which can even use a remote ASR for complex tasks.

1 Introduction

Several years ago to deploy an embedded ASR application, the main challenge was not ASR itself but the hardware, the ASR system was supposed to run on. Affordable and fast mini PCs did not exist yet. Software-wise suitable HW usually required expensive and time consuming porting and testing. Significantly increased performance of affordable Linux based mini PCs allowed to design and implement a simple, reliable, inexpensive, and especially today, widely usable speech recognition systems for many different applications. Pocket-sized computers can today easily outperform normal desktop PCs from some years ago and unified development environments allows very fast porting to a new target hardware.

Another limiting factor for many ASR applications was availability. ASR systems were usually connected with some device like a PC, a car or installed access point for ASR with microphone and button. For always available ASR in a building a theoretical option was a set of microphones installed everywhere but in a real life such a solution is not acceptable.

Changes in the mobile phones world several years ago helped to face this problem. Most of the middle-class and state-of-the-art mobile devices today are equipped also with wireless network support. With such a mobile device a user does not need to access a microphone attached to a PC or some wall. Via wireless network the mobile phones can easily access also local ASR systems. So

P. Hliněný et al. (Eds.): MEMICS 2014, LNCS 8934, pp. 16–29, 2014.
DOI: 10.1007/978-3-319-14896-0_2

mobile phones are acting as a remote microphone. Besides that, a change in user behavior has occurred: The mobile phones are already accepted and being used through all age groups [5].

Mobiles phones today are powerful enough to run even system for automatic speech recognition. Unfortunately, the variety of mobile devices prevents to design a low cost speech recognition software running on all available mobile phones reliably. Indeed it is much easier to design a simple application which is just recording the speech signal. Applying a client–server approach, the recorded speech signal from the mobile device can be send to a recognition server for processing. As already pointed out such a "server" can today also be a cheap device which has a similar size as the mobile phone itself and can be installed wherever ASR application is needed.

Today, speech recognition on a mobile phone or on a PC is a common feature. Despite the fact that it is mainly a remote service the latency and the accuracy is acceptable for given task. Why then use a local ASR system? We will answer this question in Section 2. In Section 3 and Section 4 we show two examples of local ASR systems with focus to aspects described in Section 2. In Section 5 a brief summary is provided.

2 Local Versus Remote ASR Systems

Free speech recognition services are easy to use today. Their integration into not speech enabled application is usually not very complicated and that is the reason why they are experimentally used also in applications clearly inappropriate for them.

There are two main reasons why such a service is not suitable for many applications with embedded ASR:

1. Free services usually do not have a specific usage domain. It means in the background is large vocabulary LM based system. Such a system can recognize everything and therefore it can be integrated into any application but the larger the vocabulary, the lower the accuracy. If we try to use such a service in an application with very limited vocabulary size (e. g. 100 words), then the accuracy will be significantly worse in contrast to ASR system with 100 words vocabulary. In case that instead of LM it is possible to use a grammar the result will be even better.
   Another disadvantage of LM against grammar for embedded ASR applications is the need of some semantic interpreter. Well written grammars with semantic tags will outperform comparable LM based system not only in case of accuracy but also in case of latency and resources required (which can be very important as we will show in the following sections).
   Despite the fact that grammar based systems can be faster and more accurate, they have also some drawbacks. To design a simple grammar is a simple task. To design a good and robust grammar is a complicated and time consuming task. Another problem are out of the grammar (OOG) utterances. A grammar based system covering just a wanted domain will always recognize

a valid sentence. Perhaps with very low confidence score but that's not very reliable rejection parameter. A low confidence score will even assigned to not OOG utterance in a very noisy environment (car driving at a high speed, people talking in the background, ...). Solution for OOG in the grammar based system can be some simple garbage model as showed in Section 4 but it makes the entire design even more complex.

2. Latency. There are 2 main sources of latency. The first one depends on vocabulary size, the used technology and the service setting. It is latency generated by the recognition system itself. More accurate or larger vocabulary usually results in increased latency. For some applications, real time processing is critical.
   The second latency source is caused by communication with a remote service. Such a latency can vary between 100ms and several seconds. In case we want to use it while being mobile (e.,g. from within a moving car) it becomes necessary to handle the case that the service is sometimes unavailable.
   We tested the latency for one of the popular freely available ASR services for commands covered by grammars used in Section 3 and Section 4. The best latency was about 1.5 second, but 4 seconds latency was nothing special. As we will show later, such a result is for the demonstrated real time applications not acceptable. It is necessary to note that for the testing we used a good Internet connection. On a mobile network in a moving car we expect even worse numbers.

In the following two sections we show two real applications with a local embedded ASR system and point out particular aspect of local grammar based system in contrast to remote LM based ASR.

3 Voice Enabled House Control

In this example we present a user interface for controlling home equipment such as lights, blinds or heating via speech. The question of how to provide the user with an easy to accept and easy to use interface/device is in the research area still going on. Some suggest the TV as a device that is readily available and accepted by people [4]. But it has the drawback that it is not mobile and it does not allow for a speech interface, which has emerged as a preferred input method. [4] stated that such a systems had the following requirements:

- light weighted
- simple and intuitive to use
- adaptable to physical and psychological changes
- offers various input methods like speech and touchscreens
- reliable

Therefore, we propose a speech interface for controlling home devices that runs on mobile phones. The mobile phone addresses several of the previously mentioned requirements in that it is light weighted, simple and intuitive to use.

Our user interface runs on the mobile phone as an additional application that allows the user to interact with their home devices. The microphone is only activated as the respective button is pushed, which addresses another issue: privacy [1]. In environments where microphones are set to always-listening modes this is a major issue, as the microphones are constantly recording. This is avoided by giving the user the control over the microphone.

The privacy is important issue in case of remote ASR system used for the house control application even if users have control over the microphone activity. They are not happy that someone else has possibly overview about their activities in their house.

So called intelligent or automated houses today are equipped by default with a central control system. Such a system is able to control and monitor many devices, like lights, shutters, doors, the heating and others. They are usually based on KNX/EIB or similar technology. The control of such a system is usually done with switches similar to those in "normal" houses. Beside, there is also a graphical user interface which allows the same functionality as standard switches but also opens the door to more advanced control and monitoring features.

Such a graphical user interface (GUI) is mostly integrated in to the wall at some fixed place — for example right beside the entrance of a house. It can also be accessible with a personal computer or via some kind of tablet PC, which allows usage from almost anywhere. However, the tablet PC is not being carried all the time with the user and can still be relatively heavy for a disabled or elderly users. If they do not have a simplified user interface, they can not be considered as user friendly for elderly people even despite individual adaptation to the user.

3.1 Overall Design

Such hardware equipment allows to make very quickly any intelligent house voice enabled. The entire architecture is shown in Figure 1. The user says the voice command in to a mobile phone. The mobile phone send it to the "server" using the available wireless network. The server will process the speech signal. After the recognition, the result is interpreted to generate the proper command for the house and also sent back to the mobile phone for visual feedback. The final command is sent to the KNX/EIB network via an interface. The entire system is working in real time and the action derived from the speech command takes place immediately. The latency of the system is below 300 ms. Such a latency is by a common user described as instant reaction.

As mentioned above in case of remote ASR service, the best latency is about 1.5 second. Such a latency can be still accepted by users. However 1.5 second was the best time. Response time about 5 second in case of lights is already not acceptable.

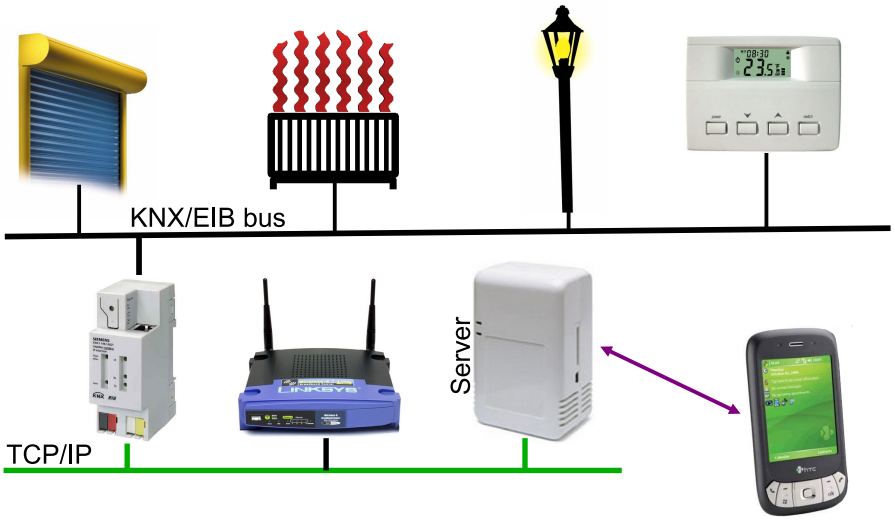

Fig. 1. Voice enabled house control architecture

3.2 Speech Recognition

Speech recognition under the technical conditions described above and controlling the utilities in an intelligent house have two important and positive features which results in high reliability of entire system:

1. The recorded speech signal has a very good quality. The mobile phone is acting as a close-talk microphone. In general, mobile phones have very good audio input hardware in contrast to many other hand-held devices where audio input is designed only as an optional feature.
2. The set of the commands for the house control is relatively small. The number of controlled utilities in average house is usually around 50. For this reason the speech recognition system can be grammar based and still very robust.

The grammar based recognition system obviously requires designing a grammar. Since each house is different, each house needs also an individual grammar. Fortunately, the group of the devices (lights, shutters, heating, ...) as well as group of available commands (switch on/off, up/down, dim, ...) is relatively small. Therefore we were able to design a fixed grammar, where during the adaptation for a particular house it is "just" necessary to add the existing devices with their real names (Peter's room, garden light, ...).

All the changes necessary for one specific house can be done on the "server". The mobile phone is running a universal speech recording software and can be used in any house where such a server based recognizer is installed.

The exemplified grammar in Figure 2 accepts for example the following commands:

```
[$prefix] $loc_garten $actionSchalten
   {out.device=rules.loc_garten.device; out.action=rules.actionSchalten.action;}
 | ([bitte] [$prefixMach]|[$prefixMach] [bitte]) $loc_garten $actionSimple
   {out.device=rules.loc_garten.device; out.action=rules.actionSimple.action;}
 | [$prefixMach] $loc_garten $actionSimple [bitte]
   {out.device=rules.loc_garten.device; out.action=rules.actionSimple.action;}
 ;
$loc_garten = ($lampe (im | in dem) Garten | [die] Gartenlampe)
   {out.device="L_Garten";}
 ;
$lampe = [das] Licht | [die] Beleuchtung
 ;
$prefix = (Wuerden Sie|Koennten Sie|Wuerdest du|Koenntest du)
[bitte]
 ;
$prefixMach = Mache|Mach | Machen Sie | drehe|dreh |
schalte|schalt
 ;
$actionSimple = (an|ein) {out.action="ON";}
 | aus {out.action="OFF";}
 ;
$actionSchalten = (einschalten|anmachen|anschalten)
{out.action="ON";}
 | (ausschalten|ausmachen) {out.action="OFF";}
 ;
```

Fig. 2. Example of a simple grammar for switching a garden light

- *"Könntest du bitte die Beleuchtung im Garten einschalten?"* (Would you please turn on the light in the garden?)
- *"Das Licht im Garten an"* (Light in the garden on)

3.3 Experiments

To test and evaluate the implemented solution we installed the entire system into real houses. After adaptation to the house environment, as described above, the system was passed to the householder for real usage. The users were not informed about the available commands. They were asked to talk to the system as they wish.

After one month we downloaded all speech commands, which were saved with the householder's consent, and transcribed them. In the Evaluation, we did not focus on the speech recognition accuracy, but on the action accuracy. For example if a user said: "*Die Beleuchtung in der Küche einschalten*" and the system recognized: "*Licht in der Küche einschalten*", then from a recognition point of view it is incorrect, but from an action accuracy point of view it is correct, as the same action would be triggered. We also analyzed out of the grammar sentences to improve the grammar to be able to cover bigger variety of utterances. In Table 1 are results for out of the grammar utterances, sentence accuracy and action accuracy for an evaluation period of 1 month with 4 different users depicted.

Table 1. Out of grammar utterances, sentence and action accuracy for evaluation period one month and four different users

Out of grammar utterances	14.93 %
Sentence accuracy	55.56 %
Action accuracy	91.23 %

The result for OOG utterances is high, but is caused by the fact, that users did not get any initial instructions. A closer look at the OOG utterances distribution in time, we can clearly observe, that most of them appear shortly after system installation. For more detailed results a longer evaluation period is needed. For sentence and action accuracy, out of the grammar utterances were removed from evaluation pool.

On the first look, 55.56 % sentence accuracy may seem very small, but it resulted in a 91.23 % action accuracy. We analyzed the recognition errors and most of the errors in prepositions like "im" or "in" or incorrectly recognized articles. Such errors are not influencing the action accuracy rate and are mostly not noticed by the user. It is also important to note, that almost 30 % of utterances were spoken by a non native speakers. Recognition errors that resulted in faulty actions usually lead to the user to retry.

Besides measuring accuracy, we asked the householder about their personal satisfaction with a free-form questionnaire. In all cases the reported satisfaction can be summarized as very high.

4 In–Car ASR for Secondary Functions

The usage of ASR systems in todays cars can be classified in two distinct classes: On the one hand there are integrated ASR systems, which control basic comfort functions like air conditioning, radio, or navigation system, e. g. to enter the address. On the other hand, todays upper class cars are utilizing speech recognition system running on a server which is accessed through the Internet. This allows for more complex tasks, e. g. supporting inquiries for weather or traffic information.

Irrespective of the used ASR technology, in general the set of controlled in–car devices and functions does not expand to the secondary functions (e. g. lights or windscreen wipers). The driver can reach those without having to stop focusing on the driving process itself. Pressing a switch is in general, faster than to use a spoken command for such a task. However, controlling comfort functions is a more complicated process. Complex tasks like music selection require a significant amount of the driver's attention. Therefore, the driver benefits from controlling these functions by voice. In cases where the driver has to use a joystick instead of a steering wheel, e. g. due to a disability, controlling the secondary functions takes significant additional effort as well. Therefore, it makes sense to expand the voice control to include also the secondary functions. The requirements for controlling secondary and comfort functions differ: On one hand a

reliable, real time ASR system with a safety model for incorrectly recognized commands is required for secondary functionalities. On the other hand controlling the comfort functions by voice, does not require real time ASR. Also, a mis-recognized comfort function does not directly influence safety.

In this example we describe our effort towards the implementation of a hybrid ASR system. A real time, grammar based embedded recognizer is used to recognize secondary functions commands directly in the car. A remote large vocabulary, LM based recognizer connected via the Internet is used for advanced comfort functionality. We investigate different methods for dynamically switch between those recognizers, which is an important step towards reaching the aforementioned goals.

4.1 Secondary and Comfort Functions

We define 3 classes of functions available in a car. They differ in terms of availability, simplicity of usage and required promptness of the reaction.

1. *Secondary functions:* Obligatory functionality of each car which does not belong to the primary functions (accelerator, breaks, steering wheel, ...). Examples are the different kind of lights, car horn or windscreen wipers. They are easily accessible and intuitively to operate. The reaction time of all these devices is instant and reliability is very high.
2. *Basic comfort functions:* Optional equipment of a car related to driving comfort, e. g. air conditioning or radio. They are usually easily accessible but not always intuitively to operate. As before, the reaction time is instant. Malfunctioning is not significantly influencing car usability.
3. *Advanced comfort functions:* Optional equipment of a car related to driving comfort, e. g. navigation system or traffic information systems. In general, they are rather complex to operate and the reaction time is not instant. Some of these functions require Internet access. Malfunctioning affects only the comfort of the driver.

Secondary functions are easily accessible in any car and there is seemingly no need to use voice control. However, the situation is fundamentally different in cars modified to be used by disabled driver. Depending on the level of disability, controlling secondary functions with ordinary control levers may vary from easy to impossible. In the latter situation, speech recognition might be a more natural way to control the secondary functions of a car.

4.2 Hybrid Speech Recognition

Because of the different requirements for the aforementioned in–car functions, it is difficult to use a single ASR system. For the secondary and basic comfort functions it is necessary to use a real time local ASR system with very high recognition accuracy. This is achieved by a small vocabulary grammar based system directly integrated into the car. The advanced comfort functions often

require a large vocabulary, but do not require as high accuracy and low latency as ASR for the secondary functions. We are using a LM based recognition server accessed through the Internet to provide this functionality. Finally, we designed a system which dynamically switches between the two recognition systems to provide a uniform interface to the user.

In the literature the term *Hybrid Speech Recognition* is used to describe a combination of HMM and ANN based recognizers. In this paper however, we use it to refer to the combination of a grammar based, real time recognizer with a remote server based, large vocabulary recognizer.

ASR for Secondary and Basic Comfort Functions. Embedded recognizers were originally designed to run on significantly slower hardware than available today. Therefore, in case of a small grammar the real time requirement is easily satisfied. The main challenge for such a system is to meet the very low error rate requirements. An incorrect recognition can trigger an unwanted action, which, in a certain ill–timed moment, can lead to dangerous situations, e. g. switching off the lights during the night or switching on the opposite turning signal. Therefore, a safety model in case of an incorrect recognition is needed.

We are using commercially available embedded recognizers. To run the recognizer we used the same platform as in House Control case. We were focusing mainly on grammar and application design to achieve maximal accuracy and reliability. Usually if the grammar offers a big variety of commands the error rate of the recognition increases. Therefore, we tried to minimize the grammar size and avoid acoustic similarities between the commands. As there are many ways to toggle specific devices, we focused on the most common short and long forms. For instance, for turning on the high beams the short form is "*Fernlicht an*" whereas the long form is "*Das Fernlicht einschalten*" [1]. The vocabulary size of the resulting grammars is only around 30 words.

Among the devices controlled by the embedded recognizer are: low beams, high beams, turning indicators, light horn, windscreen wipers. It is possible to switch them on, off and in case of turning indicators to let them on for few seconds only, e. g. to indicate overtaking.

The system is operating in *Push-to-Talk* (P2T) mode, which means that the system is only listening while a button is pressed. The *Push-to-Activate* (P2A) mode, in which the user only pushes the button once to indicate the start of the utterance, could be easier to use. However, we decided for the P2T system for accuracy reasons. Especially at high speeds the automatic end–pointing needed in the P2A system poses a problem due to the environmental noise. The second reason for P2T mode is the latency.

Irrespective of the activation mode, the button used is serving also safety purposes. If the user presses the button again shortly after the recognition finished, he cancels the initiated action. Such a behavior should avoid unwanted situations caused by incorrect speech recognition and consecutive actions.

[1] German terms to switch on the high beams.

ASR for Advanced Comfort Functions. In order to provide the user with the comfort functions as defined in Section 4.1, the speech recognition system must be able to deal with a large vocabulary. Therefore, it is no longer feasible to use a grammar based recognition system. We decided to use a remote server based, large vocabulary speech recognition system. It is located in a computing center and consequently requires an in–car Internet connection to be available.

Regarding the recognition time, there are two considerations to take into account: On one hand, in case of accessing the advanced comfort functions it is no longer necessary to provide the user with recognition results in real time. On the other hand, it is also important that processing is not taking too long as the driver gets distracted from driving when the system is not working as he expects to, Given that the audio data needs to be transferred to the server which in turn sends back the recognition result using a possibly slow and unreliable mobile Internet connection, it was necessary to build a robust system which can handle outages in a non-disruptive way.

In order to decrease the recognition time, the service uses a custom network protocol to transfer the audio data in small chunks. The protocol allows the server to send back partial results as soon as they are available. Optimizing the server–side processing of the received audio signal allows to further decrease the perceived decoding time. Using this technique, we were able to reduce the perceived recognition time factor from around 3 down to around 1. The *perceived recognition time* specifies the time the user perceives as waiting time from finishing to speak until the system reacts to his input. The *actual recognition time* can differ, mainly due to the time needed to transfer the data to the server.

The recognition system we are using is working with a language model with a vocabulary size of over 1 million words, specifically tailored for mobile search and dictation applications. The server based system is designed to be highly scalable and can serve many clients at the same time without performance degradation.

Which One to Use? The audio signal is always processed by the in–car recognition system. A control application has to decide if the command was aimed at the secondary or basic comfort functionality or whether it is part of the advanced comfort functions. We evaluated 3 different approaches on how to distinguish between them:

1. *Confidence score:* Only the confidence score of the recognized utterance is taken into account. If the score is below a certain threshold, the audio signal is sent to the server based recognizer.
2. *Out of grammar model:* If the recognition result is tagged as OOG, the audio signal is sent to the server based recognizer. The confidence score is not taken into account.
3. *OOG model with trigger word:* As the previous method, but a special key word has to precede the "out of grammar" part.

In all of these cases, at first the in–car recognizer is trying to recognize the command. The final decision is taken based on the recognition result, the confidence

score or the length of the utterance. If the decision algorithm decides that the utterance has to be sent to the server based recognizer, the application informs the user about it and waits for a reply from the server.

4.3 Evaluation

The evaluation is split into two major aspects. The first aspect is to examine the speech recognition accuracy for different grammars and noise levels. The second aspect is evaluating the switching between the local and the remote recognizer. In order to evaluate our system we recorded a test set. For data collection the P2T mode was used and the microphone was at a distance of 20−30 cm to the speaker. The recorded data consists of 10 speakers (4 females and 6 males) of which 2 were non-native German speakers. For each we recorded 2×30 commands, containing

- 10 long commands for controlling secondary functions (*den Blinker links ausschalten, die Lichthupe einschalten, ...*),
- 10 short commands for controlling secondary functions (*Blinker links an, Lichthupe, ...*),
- 5 commands controlling comfort functions with a trigger word (*Komfortfunktion: Wettervorhersage für Heidelberg, Komfortfunktion: Radio: SWR3 wählen, ...*), and
- 5 commands controlling comfort functions without a trigger word (*Wettervorhersage für Heidelberg, Radio: SWR3 wählen, ...*).

The recording took place in 2 different environments: A quiet office environment and a noisy environment with in–car noise up to 80 dB, responsible for low SNR and the Lombard effect during the recording.

Speech Recognition. For the recognition accuracy test we created two different grammars. The first grammar is covering only the long forms of the commands and was designed to be used only with the first 10 test sentences recorded by each speaker. The second grammar is covering all commands for the secondary functions. The second one was used for all recorded commands to examine whether the error rate is getting worse with bigger command variety in the recognition grammar as expected. However, more important than the speech recognition accuracy is the action accuracy. Therefore we examined action accuracy as well as recognition accuracy.

In Table 2 the results for the sentence accuracy and the action accuracy obtained on the test set are shown. From the speech recognition point of view the most important results are the sentence accuracy (SA) and sentence error rate (SER). It is difficult to decide which combination of grammar and set of commands to use based on these results alone. In the quiet environment the short form commands with the full grammar give the best accuracy, whereas in the noisy environments the long forms with the reduced grammars give the best results.

Table 2. Speech recognition and action accuracy (SER – Sentence Error Rate, SA – Sentence Accuracy, AER – Action Error Rate, AA – Action Accuracy, ASCF – Average Sentence Confidence Score)

	SER	**SA**	**AER**	**AA**	**ASCF**
			Quiet environment		
Long form - reduced grammar	2 %	84 %	2 %	94 %	84.51 %
Long form - full grammar	15 %	80 %	1 %	94 %	84.56 %
Short form - full grammar	9 %	91 %	3 %	97 %	84.01 %
			Noisy environment		
Long form - reduced grammar	0 %	94 %	0 %	84 %	81.88 %
Long form - full grammar	13 %	86 %	1 %	98 %	81.38 %
Short form - full grammar	11 %	88 %	6 %	93 %	77.36 %

Taking the action accuracy (AA) and more importantly the action error rate (AER) into account, Table 2 gives a better indication which is the safest grammar and commands combination. The smallest AER and biggest AA are always achieving using the long form of commands. Whether the grammar should also contain the short forms is subject to practical testing.

The table shows also the average sentence confidence scores[2]. We did not take into account the confidence score during the evaluation. However, using also such an information is an option how to further eliminate incorrect actions caused by an incorrect recognition result. On the other side the result rejection based on the confidence score will decreased the action accuracy. The number of commands from the recognition test with confidence score below 50% was 5. In 4 of these 5 cases the recognition was incorrect. Therefore, if we used a minimum sentence confidence score for the secondary functions of 50%, it would further reduce SER or AER but AA as well.

Later testing in real car where users were properly instructed about the system and later were driving without any interruptions resulted in average AA 96%. Average SA for given test was about 70%. The real system was rejecting all the command with the confidence score below 50%.

Speech Recognizer Selection. The recognizer selection tests included all three approaches described in Section 4.2. For the confidence score approach we reused the grammars used for the tests in Section 4.3. With those grammars we tried to recognize the recorded commands aimed at the comfort functions. Of course the recognizer produced a recognition result containing a sentence from the grammar. But now the sentence confidence score is taken into account as well. Therefore, we examined the maximum score a sentence for a comfort function would gain, which are listed in Table 3. Comparing these values with

[2] Confidence score of a particular recognizer was scaled into to the range 0 to 100.

Table 3. Maximal sentence confidence score for the comfort function commands with the secondary function grammars

	With trigger word	**Without trigger word**
	Quiet environment	
Reduced grammar	36 %	59 %
Full grammar	42 %	60 %
	Noisy environment	
Reduced grammar	42 %	46 %
Full grammar	42 %	61 %

Table 4. Out of grammar (OOG) recognized for secondary function commands

	Quiet env.	**Noisy env.**
OOG without trigger word	76 %	84 %
OOG with trigger word	0 %	0 %

the sentence confidence scores reported in Table 2, in all cases we observe a satisfactory difference. The lowest confidence scores were achieved for commands containing a trigger word. The best result was achieved with the combination of using such a trigger word and the grammar containing only the long forms.

For the garbage based experiments, we modified the recognition grammar to include also an out of grammar (OOG) model. In the experiment with garbage preceded by a trigger word a command "<`Trigger word`> `OOG;`" was added. In the other experiment just the command "`OOG;`" was added. We were observing how many times the result "`OOG`" appeared among the recognized commands for secondary functions and how many times "`OOG`" did not appear among the comfort functions commands.

Table 4 shows how often "`OOG`" was returned when feeding secondary function commands into the speech recognition engine. We did the experiment with and without the trigger word "Komfortfunktion" which is not part of the remaining grammar. The results indicate, that for a reliable separation of secondary and comfort functions, the usage of some kind of trigger word is necessary. In the following experiment we used the grammar containing the trigger word and used the comfort function commands as input for the recognizer. In nearly all cases (98 % in quiet, 100 % in noisy environment) the recognizer returned the "`OOG`" indicator.

In case of the comfort functions the error rate, i. e. cases in which the output should be "`OOG`" but was not, is more important than the accuracy. A comfort function command which is accepted by a secondary function grammar could trigger an unwanted action on the secondary functionality in the car. The error rate measured in quiet and noisy environment was 0 %. Consequently, the results

are confirming the previous indication, that the usage of an adequate trigger word is a reliable way to determine which recognizer to use.

5 Summary

In this paper we described nowadays changes in design and development of embedded ASR applications. We started with the description of changes in availability of ASR services and deployment possibility of small local as well as remote embedded ASR systems. On two real time application we demonstrated the significance of local grammar based systems and showed also example how local and remote ASR system can be combined in to one application.

Given examples also implied a new areas where ASR can be used today. Hardware evolution and changes in users behavior are opening doors to areas hardly imaginable 10 year ago.

References

1. Caine, K.E., Fisk, A.D., Rogers, W.A.: Benefits and privacy concerns of a home equippend with a visual sensing system: a perspective from older adults. In: Proc. of the Human Factors and Ergonomics Society 50th Annual Meeting (2006)
2. Ivanecký, J., Mehlhase, S., Mieskes, M.: An intelligent house control using speech recognition with integrated localization. In: Wichert, R., Eberhardt, B. (eds.) Ambient Assisted Living. Non-series, vol. 63, pp. 51–62. Springer, Heidelberg (2011)
3. Ivanecký, J., Mehlhase, S.: An in-car speech recognition system for disabled drivers. In: Sojka, P., Horák, A., Kopeček, I., Pala, K. (eds.) TSD 2012. LNCS, vol. 7499, pp. 505–512. Springer, Heidelberg (2012)
4. Lienert, K., Spittel, S., Stiller, C., Roß, F., Ament, C., Lutherdt, S., Witte, H.: Seniorenbefragung zum Assistenzsystem WEITBLICK - Ergebnisse einer Bedarfsanalyse. In: Proc. of the Third German AAL Conference (2010)
5. Van Bronswijk, J.E.M.H., Kearns, W.D., Normie, L.R.: ICT infrastructures inthe aging society. International Journal of the Fundamental Aspects of Technology to Serve the Ageing Society – Gerontechnology **6**(3) (2007)

http://www.springer.com/978-3-319-14895-3

Mathematical and Engineering Methods in Computer Science
9th International Doctoral Workshop, MEMICS 2014, Telč, Czech Republic, October 17--19, 2014, Revised Selected Papers
Hliněný, P.; Dvořák, Z.; Jaroš, J.; Kofroň, J.; Kořenek, J.; Matula, P.; Pala, K. (Eds.)
2014, XI, 159 p. 50 illus., Softcover
ISBN: 978-3-319-14895-3