ON THE IMPLEMENTATION OF ASR ALGORITHMS FOR HAND–HELD WIRELESS MOBILE DEVICES

R. C. Rose[1], *S. Parthasarathy*[1], *B. Gajic*[2], *A. E. Rosenberg*[1], *S. Narayanan*[3]
[1]*AT&T Labs – Research, Florham Park, NJ 07932*
[2]*Norwegian University of Science and Technology (NTNU), Trondheim, Norway*
[3]*Signal and Image Proc. Institute, USC, Los Angeles, CA 90089*

ABSTRACT

This paper is concerned with the implementation of automatic speech recognition (ASR) based services on wireless mobile devices. Techniques are investigated for improving the performance of ASR systems in the context of the devices themselves, the environments that they are used in, and the networks they are connected to. A set of ASR tasks and ASR system architectures that are applicable to a wide range of simple mobile devices is presented. A prototype ASR based service is defined and the implementation of the service on a wireless mobile device is described. A database of speech utterances was collected from a population of fifty users interacting with this prototype service in multiple environments. An experimental study was performed where model compensation procedures for improving acoustic robustness and lattice rescoring procedures for reducing task perplexity were evaluated on this speech corpus.

1 INTRODUCTION

The widespread interest in ASR on hand–held mobile devices is due to the perception that speech input provides considerable added value given the limited input modalities that can be supported on these devices. A variety of ASR tasks relating to portable devices have been envisioned ranging from portable dictation systems to data entry systems implemented on specialized multi–media devices [1, 2]. Furthermore, a variety of ASR system architectures have been implemented ranging from server based implementations accessed by the device over the wireless network to recognizers embedded in the local processor associated with the device [5]. This paper investigates ASR issues in the context of a limited set of tasks and architectural alternatives for ASR services in a wireless mobile environment. Specifically, we are interested in tasks that provide functionality which is equivalent to form filling applications that are currently available only on an internet browser running on a desk-top workstation with a keyboard and a mouse. We are interested in network based ASR architectures, and have implemented a prototype system based on that framework.

The choice of task and system architecture is dictated by several issues. The first issue is the inherent limitations of the devices and the variety of conditions under which they are used. There will always be fundamental trade-offs between computing power and battery life, and the available bandwidth will always be limited by the existing telecommunications infrastructure. A second set of issues are security and consistency considerations for large real-time databases. These limitations make it difficult to build recognition networks on the mobile device that are associated with very large databases, as might be the case, for example, with a directory retrieval application with hundreds of thousands of names. A third issue dictating our choice of tasks and system architecture is applicability to a large range of simple hand–held devices. It is assumed that these devices would include voice input in combination with pen or push-button input and a limited display. Our focus is primarily on voice–enabling the widest possible variety of hand–held devices rather than to develop specialized multi–media portable systems.

The general "form–filling" class of tasks considered here will be described in Section 2. Issues relating to ASR system architectures in a wireless network will be discussed in Section 3. As part of this discussion, the operation and the architecture of our prototype ASR based application running on a hand–held device will also be described.

The simple task and prototype implementation serve as a platform for investigating ASR problems that are specific to wireless mobile devices. Some of these problems are discussed in Section 4, where two approaches for improving ASR performance on this task are described. Acoustic hidden Markov model (HMM) compensation procedures for dealing with difficult acoustic environments are described in Section 4 and evaluated on this task in Section 5. Techniques for reducing task perplexity by rescoring lattices from utterances taken from multiple parts of a dialog are also described and evaluated. Section 5 provides a description of the speech corpus which was collected using this prototype system. Its use as an evaluation platform for an experimental study involving the above techniques is presented.

2 TASK DOMAIN

2.1 Form–Filling Applications

The interest in this work is to investigate ASR problems in the context of a class of services that will be most widely applicable to a broad family of simple wireless devices. In particular, simple "form–filling" applications that are easily implemented using an internet browser on a desk–top workstation are considered. A good example of this class of services is navigation of the on–screen dialog that is associated with a typical travel services web page. These on–screen dialogs involve a user clicking on a field in the display, typing a value for that field, and continuing to fill in fields until enough information has been entered to form the desired database query. While the need for keyboard input would make it difficult to extend this functionality to a standard mobile device, this

class of services would be straight–forward to implement on a small device that employs a combination of speech and pen input along with a simple display. The design of speech based dialog provides input confirmation strategies and guides user behavior to fill in fields associated with a semantic frame is much simpler for this scenario than for a voice–only dialog systems. Even a very simple display can provide much of the context for the dialog, and speech input would be easier than handwriting input or selecting fields from long menus.

2.2 A Prototype ASR Task

The ASR task used in this work is based on a 3700 name ATT Labs employee directory access task where users click on individual data fields for a database entry, speak the entry for that field, and see the recognized result displayed in the field as confirmation. An example of the display associated with this application is shown in Figure 1. The data fields include first name, last name, location, and telephone number. Once enough fields have been entered to uniquely disambiguate the desired directory entry from all other entries in the database, all available information for that user is displayed in the data window at the bottom of Figure 1. The speech corpus described in Section 5 was collected from users who were interacting with the display in Figure 1.

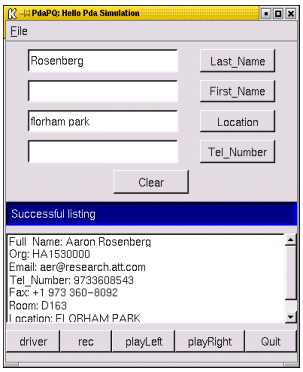

Figure 1: Display for multi–modal directory retrieval service.

3 SYSTEM ARCHITECTURE

3.1 Server Based ASR

The block diagram in Figure 2 is a simplified description of how the ASR based service described in Section 2 would be implemented in a wireless telephony context. The hand–held device runs a thin client that implements a simple application. The application interprets input generated on the device and communicates with the ASR and dialog servers via client/server protocols. The speech recognizer and the dialog manager exist as servers in the network. Speech is sent from the application to the ASR server either as coded ASR features over the ASR client/server protocol or over a voice channel. The ASR server provides the result to the application either as a single string or a word lattice. The application running on the hand–held device presents the dialog manager with new input information from the device which corresponds to the values associated with fields that have been entered by the user. With the newly updated input fields and the application database as input, the dialog manager output is used by the application to either display directory information to the user or to prompt the user for additional information.

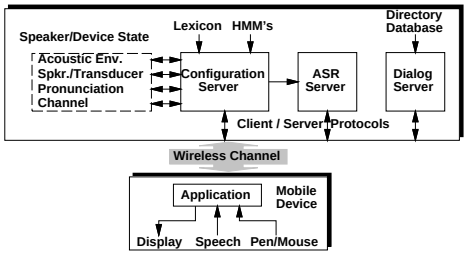

Figure 2: Architecture for implementation of ASR based service on hand-held wireless mobile device.

The application can instruct the ASR server to update acoustic HMMs and pronunciation models to reflect a given device, user, or acoustic environment. It can also select the proper recognition network to associate with a given database field. Acoustic parameters associated with the speaker, device, and acoustic environment can be continually updated through a "configuration server" and stored in the network. These parameters can then be applied to compensating and adapting the speaker independent HMMs associated with the network based ASR server. The techniques discussed in Section 4 are all implemented in a manner that is consistent with this scenario.

3.2 Prototype Implementations

The prototype versions of the general system architecture shown in Figure 2 were implemented in the context of an 11 Mbps IEEE 802.11 wireless local area network. Network bandwidth and channel distortions were not an issue in the prototype implementation. The application runs on a Linux based lap–top PC. It is being ported to a Compaq iPAQ hand–held device running Linux. The ASR server is implemented using the AT&T Watson speech recognizer [9] and the dialog server is implemented using the Chronus dialog manager [8].

4 ASR ISSUES ON MOBILE DEVICES

There are many issues for mobile devices that affect the actual ASR algorithms that are implemented in this domain. The first set of issues relates to the variety of conditions under which these devices are being used. The mobility afforded by wireless connectivity implies a wider variety of acoustic environments than those that exist for wire-line telephone or desk–top computer applications. A second set of issues is related to the long–term use of the device by a single user. The fact that a device can be "personalized" to a given user represents an opportunity to acquire representations of speaker, environment, and transducer variability through the normal use of the device. A third set of issues relates to the nature of the ASR based services that are likely to be used on real commercial devices. While the "form–filling" scenario described in Section 2 seems like a simple class of tasks, the vocabularies involved can be large and statistical language models are not applicable when the task is to recognize 1 out of N equally likely options in a field. These issues are addressed in the following ways.

4.1 Compensation for Acoustic Variability

Techniques for compensating speaker independent acoustic HMMs simultaneously for environmental noise and for speaker/transducer mismatch have been applied to this task [3]. These techniques are based on iterative application of parallel model combination (PMC) of speech and noise models and maximum likelihood (ML) based linear

model compensation procedures [3, 6, 4]. It is assumed that the ASR system in Figure 2 relies on a speaker independent HMM λ_X which was trained from a sequence of observation vectors, X. It is also assumed that during recognition there is a mismatch between a corrupted sequence of speech observation vectors, Y, that is caused by speaker and transducer variability along with varying characteristics associated with the ambient acoustic noise, N. Hence, the corrupted observation vectors are modeled as $Y = WX + N$, where W is a linear transformation representing the mismatch between Y and X that is induced by the speaker/transducer mismatch.

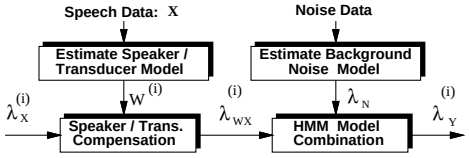

Figure 3: Procedure for simultaneous compensation of HMM with respect to ambient acoustic noise and speaker/transducer distortions.

Figure 3 summarizes an iterative procedure where at iteration i ML–based compensation of HMM model $\lambda_X^{(i)}$ is performed to obtain model $\lambda_{WX}^{(i)}$, which is then combined with estimated noise model λ_N to obtain the compensated model $\lambda_Y^{(i)}$. The performance of this procedure in reducing degradation in speech recognition word accuracy (WAC) caused by the use of far field microphones is described in Section 5.

4.2 Applying Linguistic Constraints

While the general class of tasks outlined in Section 2 seems very simple, the vocabularies involved can be large and statistical language models may not apply. Hence, although added display and input modalities may help provide additional context for the application, this class of tasks can result in a high language model perplexity which generally corresponds to high word error rates. In order to avoid these problems, linguistic constraints are applied in a lattice rescoring paradigm. The system functions as follows. First, all the queries are presented to the user simultaneously as shown in Figure 1 with a separate language model to handle the responses to each query. Second, the user selects a field, speaks the "value" of the field, and the recognizer produces a word lattice for the utterance. Third, the user continues providing responses to other queries, and the system produces a recognition lattice for each input.

Each time the ASR outputs a new lattice, a new recognition result is generated for the current input and each of the previous responses. This done in two steps. First, the lattices from each recognized field are concatenated to form a single combined lattice. Second, the concatenated lattices are re–scored using a language model that incorporates the constraints over all the fields that have been selected thus far. This process can be done in near real-time because the language models can be precomputed and the computing needed for determining the best path through the lattice is small. If the system confidence exceeds a certain level, feedback will be provided to the user. The task performance of such a system should be comparable to the situation where dynamic language models are used, in cases where that is feasible. The performance of this lattice rescoring approach is given for a simple scenario in Section 5.

5 AN EXPERIMENTAL STUDY

This section describes the results of an experimental study evaluating the effectiveness of the procedures described in Section 4. The important aspect of this study is that it represents the first time many of these techniques have been evaluated using utterances collected from speakers interacting with a real application in the actual noisy environments. Previous evaluation of similar techniques were performed by simulating noisy environments by artificially corrupting speech utterances recorded in quiet noise–free environments. The acoustic model compensation techniques are evaluated to demonstrate the potential for reducing ASR performance degradation in the context of far–field microphones, varying ambient acoustic environments, and diverse speaker populations. The lattice rescoring procedure is evaluated in order to demonstrate the potential for applying linguistic constraints over the fields in the user interface scenario associated with the task in Section 2. All of the experiments were performed using speaker independent, task-independent HMM's trained from a subset of the Wall Street Journal speech corpus recorded using a wide-band noise cancelling microphone in a high signal to noise ratio (SNR) acoustic environment [7]. Both techniques were evaluated using a speech corpus collected using the prototype system implementation described in Section 3.

5.1 Speech Corpus

A corpus of speech utterances was collected from users interacting with the application illustrated in Figure 1. The utterances were collected in both office and cafeteria environments from users speaking simultaneously through two microphones. Table 1 summarizes the utterances in the subset of this corpus that were used for the experimental study described below. The utterances used for the study were actually isolated utterances of proper names. All utterances were collected simultaneously through both close–talking (C-T) and far–field (F-F) microphones. The Sennheiser HMD410 headset with dynamic pressure gradient transducer was used for the close–talking microphone, and the Lucent Intelligent Audio SDM 1100 microphone mounted on the device at a distance of approximately 0.75 meters from the speaker was used for the far–field microphone. Each of a population of fifty subjects were asked to retrieve a different set of forty directory entries from the employee directory. The subjects were not prompted with the correct pronunciations for the fields shown in Figure 1, and the lexical pronunciations were obtained automatically using a text–to–speech system.

Information Field	Vocabulary Size	Test utterances	
		Office 48 Spkrs	Cafeteria 21 Spkrs
First Name	1884	1970	918
Last Name	2980	1963	917

Table 1: Summary of vocabulary size, number of utterances, and number of speakers for speech corpus collected in office and cafeteria environments.

5.2 Acoustic Model Compensation

The PMC and speaker/transducer compensation (TC) algorithms were evaluated in terms of their ability to minimize performance degradation between the close–talking (C-T) and far–field (F-F) microphones. Table 2 presents the performance of the model compensation algorithms

on the speech collected in the actual office and cafeteria environments through the F-F microphone. The recognition word accuracies obtained when no model compensation was performed in the office and cafeteria environments are shown in the first two columns of Table 2. The relative performance degradation between the baseline (C-T microphone) and the Uncompensated (F-F microphone) conditions is 41.7% in the office environment and 79.2% in the cafeteria environment. Looking at the relative impact of PMC and TC in the two different environments, it is clear that additive noise distortion is more dominant in the cafeteria environment then in the office environment. We also see that the combined approach showed a significant improvement beyond that achieved for either PMC or TC separately for both environments. Finally, the overall performance degradation with respect to the C–T microphone performance was reduced from 41.7 percent to 10.4 percent for the office environment, and from 79.2 percent to 39.8 percent in the cafeteria environment.

Environment	Base-Line (C-T)	Far-Field Microphone			
		Un-Comp.	PMC	TC	Combined
Office	72.2	42.1	61.0	58.9	64.7
Cafeteria	71.1	14.8	37.7	26.2	42.8

Table 2: Model Compensation word accuracy for far–field (F-F) microphone in office and cafeteria environments on the "Last Name" grammar.

5.3 Lattice combination results

The lattice re-scoring procedure was evaluated in terms of its ability to improve recognition accuracy by combining lattices obtained from utterances spoken for multiple independent fields in Figure 1. Specifically, the goal is to determine the degree to which combined recognition of the first name and last name fields improves when lattices from these two fields are combined and rescored. The experiments were performed on a subset of the test utterances shown in Table 1 that consisted of 1849 first name and last name utterances. The use of a test set that is slightly different from the test set used in Section 5.2 was mandated by the need to maintain a correspondence between the first and last names as they were spoken in the database. Only the wide–band C-T microphone utterances were used. The recognition results are reported in Table 3.

First Name	Last Name	Top-Choice Combined	Lattice	Concatenated Utterances
74.2	73.9	56.5	89.1	89.9

Table 3: Performance of the lattice combination method compared to the performance of individual fields measured in percent word accuracy

The first and second columns of Table 3 correspond to the accuracy with which first and last names respectively were recognized from independent first name and last name utterances. Recognition was performed using grammars with vocabulary sizes shown in Table 1. If the top-choice of the recognizer for the first and last names are concatenated to produce a full name, the resulting full name accuracy is significantly lower than the accuracy of the individual fields. This is shown in the column labeled "Top-Choice". The "Lattice" results are obtained using the algorithm described in Section 4.2. Finally, the column labeled "Concatenated Utterances" displays the WAC obtained by concatenating the waveforms of the first and last name utterances and using a grammar of 3701 full names from the employee database. This is the *best case* scenario in terms of performance, but incurs a maximum overhead in computation and delay. This example illustrates that lattice–rescoring of the concatenated utterances achieves WAC that is almost identical to the best-case scenario. This ability to combine input obtained independently for a variety of fields, while maintaining good performance, significantly simplifies task design.

6 SUMMARY

A prototype implementation of a multi–modal, speech recognition based service for a wireless mobile device has been presented. This implementation has served as a platform for investigating ASR problems that are specific to these devices. The baseline performance reported on this task provides a good example of how simple user interfaces but highly variable acoustic conditions can make ASR difficult. Seemingly simple user interfaces can lead to very high perplexity ASR tasks, and unencumbering microphones can exacerbate the effects of the surrounding acoustic environment. The acoustic model compensation and lattice re-scoring techniques presented here are examples of techniques that have been successfully applied to improving ASR performance. There are a variety of other problems that are important in this task domain but have not been addressed here. These include better techniques for integrating additional input and output modalities with speech, techniques for obtaining speaker-specific pronunciation models, and techniques for dealing with the vast array of robustness issues that are specific to ASR applications on mobile devices. Research on these problems will have increasing importance as voice enabled wireless devices become ubiquitous.

REFERENCES

[1] S. Ditlea. The PC goes ready-to-wear. *Spectrum Magazine*, 37(10):35–39, October 2000.

[2] S. Furui. Speech recognition technology in the ubiquitous/wearable computing environment. *Proc. Int. Conf. on Acoust., Speech, and Sig. Processing*, pages 3735–3738, June 2000.

[3] B. Gajic and R. C. Rose. Hidden Markov model environmental compensation for automatic speech recognition on hand–held mobile devices. *Proc. Int. Conf. on Spoken Language Processing*, October 2000.

[4] Mark J. F. Gales and Steve J. Young. Robust continuous speech recognition using parallel model combination. *IEEE Trans. on Speech and Audio Processing*, 4(5):352–359, September 1996.

[5] S. H. Maes, D. Chazan, G. Cohen, and R. Hoory. Conversational networking: conversational protocols for transport, coding, and control. *Proc. Int. Conf. on Spoken Language Processing*, October 2000.

[6] Yasuhiro Minami and Sadaoki Furui. Adaptation method based on HMM composition and EM algorithm. In *Proc. IEEE Intl. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, pages 327–330, May 1996.

[7] D. Paul and J. Baker. The design for the Wall Street Journal-based CSR corpus. *Proc. DARPA Speech and Natural Language Workshop*, Feb. 1992.

[8] R. Pieraccini, E. Levin, and W. Eckert. AMICA: the AT&T mixed initiative conversational architecture. *Proc. European Conf. on Speech Communications*, Sept. 1997.

[9] R. D. Sharp, E. Bocchieri, C. Castillo, S. Parthasarathy, C. Rath, M. Riley, and J. Rowland. The Watson speech recognition engine. *Proc. European Conf. on Speech Communications*, Sept. 1997.