AN EFFICIENT FRAMEWORK FOR ROBUST MOBILE SPEECH RECOGNITION SERVICES

R. C. Rose, I. Arizmendi, and S. Parthasarathy
AT&T Labs – Research, Florham Park, NJ 07932
{rose,iker,sps}@research.att.com

ABSTRACT

A distributed framework for implementing automatic speech recognition (ASR) services on wireless mobile devices is presented. The framework is shown to scale easily to support a large number of mobile users connected over a wireless network and degrade gracefully under peak loads. The importance of using robust acoustic modeling techniques is demonstrated for situations when the use of specialized acoustic transducers on the mobile devices is not practical. It is shown that unsupervised acoustic normalization and adaptation techniques can reduce speech recognition word error rate (WER) by 30 percent. It is also shown that an unsupervised paradigm for updating and applying these robust modeling algorithms can be efficiently implemented within the distributed framework.

1 INTRODUCTION

This paper describes and evaluates a distributed ASR framework for mobile ASR services. The framework is evaluated in terms of its ability to support a large number of simulated clients simultaneously using a limited set of ASR decoders. The framework currently supports directory retrieval ASR applications for users of Compaq iPAQ mobile devices over an IEEE 802.11 wireless local area network [5]. An experimental study is presented demonstrating the effect of unsupervised speaker and environment compensation algorithms in improving ASR performance when user utterances are spoken through the standard iPAQ device mounted microphone.

There are a large number of applications for mobile devices that include automatic speech recognition (ASR) as a key component of the user interface. These include mutlimodal dialog applications [3], voice form filling applications [5], and value added applications that provide short-cuts to user interface functions. Speech recognition is generally just one part of a multi-modal dialog architecture for these mobile applications whose functional components can be distributed in different ways between computing resources residing in the network and on the mobile device.

While there are a range of potential distributed ASR architectures that have been proposed for these applications, one can make qualitative arguments for when either fully embedded ASR implementations or network based implementations are most appropriate. It is generally thought that fully embedded implementations are most appropriate for value added applications like name dialing or digit dialing, largely because no network connectivity is necessary when ASR is implemented locally on the device [6]. Distributed or network based ASR implementations are considered appropriate for ASR based services that require access to large application specific databases where issues of database security and integrity make it impractical to distribute representations of the database to all devices [5]. Network based implementations also facilitate porting the application to multiple languages and multiple applications without having to affect changes to the individual devices in the network.

Acoustic variability in mobile domains is considered here to be a very important problem that distinguishes ASR in mobile domains from generic ASR domains. The main issue is that users of mobile devices will be using them in a wider variety of continuously varying acoustic environments making the expected conditions far different than one would expect in wire-line telephone or desk-top applications. However, the use of personalized devices and personalized services facilitates a new paradigm for implementing robust algorithms. Speaker, channel, and environment representations can be acquired through normal use of the device all of which can be applied to feature space and model space transformation in ASR. The feature domain speaker normalization/transformation algorithms described in Section 3 are applied and evaluated under this paradigm.

The paper is composed of two major parts. The first part, given in Section 2, will present a description of the framework along with simulations demonstrating the ability of the framework to scale to a large number of clients. The second part, given in Section 3, discusses the implementation of speaker specific feature space normalizations and transformations from user state information acquired and stored by the software framework in the network. The results of the simulations will be summarized in Section 4.

2 MOBILE ASR FRAMEWORK

Modern multi-user applications are often challenged by the need to scale to a potentially large number of users while minimizing the degradation in service response even under peak load conditions. Scaling multi–modal applications that include ASR as an input modality presents an additional hurdle as there is typically a great disparity between the number of potentially active users and a system's limited ability to provide computationally intensive ASR services. This section provides an overview of a proposed distributed speech enabling middleware (DSEM) framework that is used to efficiently implement multi–modal applications that maximize performance under normal loads and are well conditioned under peak loads. The section is comprised of two parts. First, the framework rationale and design are briefly described. The second part of the section presents an experimental study demonstrating the throughput of the framework

in the context of hundreds of simulated mobile clients simultaneously accessing a system equipped with a limited number of ASR decoders.

2.1 Description

Traditional non-ASR server implementations that assign a thread or process per client suffer from greatly degraded performance as the number of clients approaches and exceeds a server's peak capacity [7]. This degradation, typically the result of context switching and synchronization overhead, is accelerated by the high IO activity necessary to support ASR services. To combat this performance loss the proposed DSEM framework uses an event-driven, non-blocking IO model which requires only a single thread to manage a large number of concurrently connected clients. In addition, an ASR decoder cache is employed to effectively share limited decoder resources among active clients.

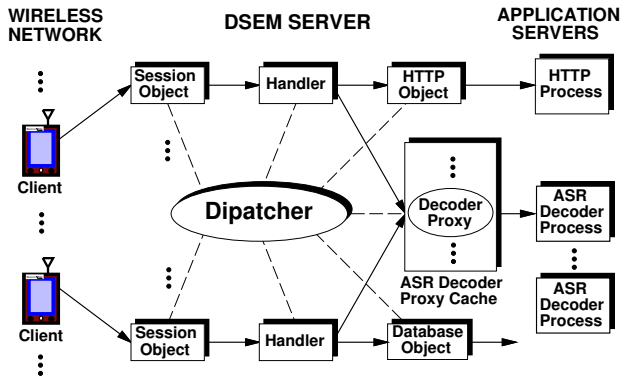

Figure 1: DSEM server framework.

The basic functional components of the framework can be introduced by way of the example illustrated by the block diagram in Figure 1. Figure 1 illustrates a typical interaction between a DSEM server and one of many clients. The interaction involves a mobile user making a voice query to an application running on the DSEM server, the decoded ASR result for the associated utterance being used to issue a HTTP query to a web server, and the result of this query being returned to the mobile client. The interaction begins with the client initiating a speech request and streaming audio to its session on the DSEM server using a custom protocol. The DSEM server dispatcher, responsible for detecting and routing all the system's IO events, notifies the session object associated with the client of the arrival of the stream. The session object serves two purposes. First, the session object is responsible for analyzing the request stream to determine the type of application–specific handler necessary to process it. Second, it is used as a repository for any client state that spans the duration of a client's session (e.g., transient acoustic information is stored here). The session can then instantiate the handler and pass it the stream for further processing.

Upon activation, the handler performs any required initialization, and attempts to acquire a decoder proxy from the decoder proxy cache. Decoder proxies act as local representations of decoder processes residing on remote dedicated compute servers. As each portion of the audio stream arrives from the client they are processed by the handler which performs cepstrum feature analysis and implements the acoustic feature space normalizations and transformations that are described in Section 3. If the handler was successful in its attempt to acquire a decoder proxy, then the computed cepstrum vectors are streamed directly to a decoding process. If it was not successful, the computed cepstrum vectors are buffered and transmitted as soon as a decoder proxy becomes available. After processing the current audio fragment, the handler returns control to the DSEM dispatcher which can then service other clients.

When the ASR decoder process obtains a recognition result, it issues a reply to its associated proxy. The dispatcher detects this event and notifies the decoder proxy so that it may read the ASR decoder's reply. The proxy then calls the handler with the recognized phrase or an optional failure code. After receiving the decoded string, the handler uses it to perform a query to a HTTP server. The prototype application implemented in this work uses this technique to retrieve employee information from AT&T's intranet site. The handler instantiates a DSEM HTTP object, issues an HTTP request and waits for a reply from the HTTP server. When the HTML response arrives, the handler can process it and send an appropriate message to the waiting mobile client.

One of the key assumptions of the above framework is that it is impractical to permanently assign an ASR decoder to a specific client. In fact, in order to support a large user population, identically configured decoder instances are stored in the cache shown in Figure 1, assigned only to an individual recognition request from a client, and then returned to the cache. As a result it is very difficult to adapt the acoustic models in the decoder itself to the user, environment, or channel associated with a particular client. The solution to this problem is to implement all acoustic modeling techniques for user configuration as feature space normalizations/transformations in the DSEM server. This issue is addressed further in Section 3.

2.2 Performance Evaluation

An experimental study was performed to demonstrate the throughput of the framework described in Section 2.1. The goal of the study was to measure both the throughput maintained by the DSEM server and the latencies that would be observed by users of the associated mobile ASR services as the number of users making simultaneous requests increases into the hundreds of users. The study was performed by having many simulated clients perform the following interaction. Each client generated a speech request to the DSEM server where acoustic feature analysis was performed, features were eventually streamed to an available ASR decoder and the decoded result was returned to the waiting client. The infrastructure used for the study included eight 1GHz Linux ASR servers with each server running four instances of the AT&T Watson ASR decoder and a single 1GHz Linux DSEM server with 256Mb of RAM.

Figure 2a illustrates the effect on response latency as the number of concurrent clients increases. Response latency was calculated as the interval in seconds between the time that the speech request was generated by the client and the time that the recognition result was returned to the client by the DSEM server. The plot in Figure 2a shows a relatively constant latency when the number of clients is less than 128 and a gracefully degrading response latency as the number of clients is increased. This increase in latency is due to the delay imposed on

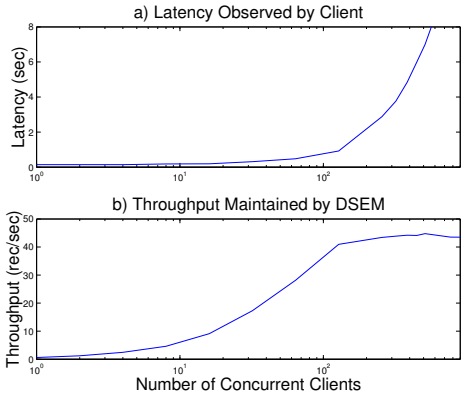

Figure 2: DSEM server performance: a) Average response latency b) Average server throughput

clients by the DSEM decoder wait queue which throttles access to available decoders and to increased pressure on the DSEM server itself.

Figure 2b illustrates the effect on server throughput as the number of concurrent clients increases. Throughput was calculated as the number of completed recognition transactions per second. The plot in Figure 2b demonstrates that throughput gradually increases until the server's peak capacity is reached at a point corresponding to 128 clients and remains constant even as the number of clients far exceeds this peak capacity.

3 ROBUST MODELING

This section describes the application of normalization and transformation algorithms in the context of the mobile ASR framework described in Section 2. These algorithms are applied to compensating utterances spoken by users of Compaq iPAQ hand-held devices. In Section 1, the notion of acquiring representations of the speaker, environment, and transducer associated with a given client from utterances spoken during the normal use of the device was discussed. The algorithms that are applied here under this paradigm include frequency warping based speaker normalization [4], constrained model adaptation (CMA) and speaker adaptive training (SAT) [2], and cepstrum and variance normalization.

There are two major constraints that are placed on acoustic compensation algorithms both by the framework described in Section 2 and by the anticipated applications described in Section 1. The first constraint is that all robust acoustic algorithms are applied in the feature space rather than by adapting or transforming the acoustic HMM model. This constraint is dictated by the dynamic assignment of decoders to individual utterances by the DSEM server making it difficult to configure the model parameters of these decoders to a particular user. The second constraint is that acoustic compensation parameters are estimated off–line from dedicated adaptation utterances rather than from the recognition utterances themselves. In addition to the fact that personalized services can be well suited to this paradigm, there are two motivations for this constraint. The first is that input utterances can be very short, sometimes single word, utterances that are spoken to fill in "voice fields" appearing on the display of the hand–held device [5]. These short utterances can be insufficient for robust parameter estimation. Second, the computational complexity associated with estimating frequency warping and CMA parameters could overwhelm the DSEM if performed at recognition time.

3.1 Algorithms

This section describes the robust acoustic compensation algorithms used for this task. They will be applied to compensating utterances spoken into a far-field device mounted microphone with respect to acoustic HMM models that were trained in a mis-matched acoustic environment. Normalization/transformation parameters are estimated using anywhere from approximately one second to one minute of speech obtained from previous utterances spoken by the user of the device.

The first technique is frequency warping based speaker normalization [4]. This was performed by selecting a single linear warping function using the adaptation utterances for a given speaker to maximize the likelihood of the adaptation speech with respect to the HMM. Then, during speech recognition for that speaker, the warping factor is retrieved and applied to scaling the frequency axis in mel-frequency cepstrum coefficient (MFCC) based feature analysis [4]. A "warped HMM" is trained by estimating optimum warping factors for all speakers in the training set and retraining the HMM model using the warped utterances.

There are several regression based adaptation algorithms that obtain maximum likelihood estimates of model transformation parameters. The techniques differ primarily in the form of the transformations. Constrained model space adaptation (CMA) is investigated here [2]. CMA estimates a model transformation $\{A, b\}$ to an HMM, λ, with means and variances μ and Σ,

$$\hat{\mu} = A\mu - b \qquad \hat{\Sigma} = A\Sigma A^T,$$

in order to maximize the likelihood of the adaptation data, X, $P(X|\lambda, A, b)$. The term "constrained" refers to the fact that the same transformation is applied to both the model means and covariances. Since the variances are transformed under CMA, it is generally considered to have some effect in compensating the HMM with respect to environmental variability as well as speaker variability.

An important implementational aspect of CMA is that this model transformation is equivalent to transforming the feature space, $\hat{x}_t = Ax_t + b$. It is applied during recognition to the 39 component feature vector composed of cepstrum observations and the appended first and second order difference cepstrum. A speaker adaptive training (SAT) HMM is trained by estimating an optimum CMA transform for each speaker in the training set and retraining the HMM model using the transformed utterances.

Cepstrum mean normalization (CMN) and cepstrum variance normalization (CVN) were also applied under a similar scenario as the algorithms described above. Normalization vectors were computed from adaptation utterances for each speaker and then used to initialize estimates of normalization vectors for each input utterance. The incorporation of additional speech data provided by this simple modification to standard cepstrum normalization procedures had a significant impact on ASR performance.

3.2 Experimental Study

The feature normalization/adaptation algorithms described in Section 3.1 were used to reduce acoustic mismatch between task independent HMM models and utterances spoken through a Compaq iPAQ hand-held device over the distributed framework described in Section 2. This section describes the scenario under which

the algorithms were evaluated, the speech database, and the experimental study.

The dataset for the study included a maximum of 400 utterances of proper names per speaker from a population of six speakers. The utterances were spoken through the device mounted microphone on the hand-held device in an office environment. Since the data collection scenario also involved interacting with the display on the hand-held device, a distance of from approximately 0.5 to 1.0 meters was maintained between the speaker and the microphone. The first 200 utterances for each speaker were used for estimating the parameters of the normalizations and transformations described in Section 3.1. After automatic endpointing, this corresponded to an average of 3.5 minutes of speech per speaker. The remaining 1200 utterances, corresponding to isolated utterances of last names, were used as a test set for the experimental study described below.

A baseline acoustic hidden Markov model (HMM) was trained from 18.4 hours of speech which corresponds to 35,900 utterances of proper names and general phrases spoken over wire-line and cellular telephone channels. After decision tree based state clustering, the models consisted of approximately 3450 states and 23,500 Gaussian densities.

The baseline WER on the above test set was found to be 41.5 percent. This can be compared to a WER of 26.1 percent obtained on the same task for a different population of speakers speaking into a close-talking noise cancelling microphone [5]. The goal of the robust compensation algorithms applied here is to close the gap between these two scenarios. It was also shown in previous work that by combining lattices obtained from utterances spoken separately in response to first name and last name fields and rescoring them with a language model that describes the constraints between those fields, a WER of 10.1 percent could be obtained [5].

Table 1 displays the results for the experimental study as the word error rate (WER) resulting from the use of each of the individual algorithms where parameters are estimated using adaptation data of varying length. Columns 2 through 5 of Table 1 correspond to the WER obtained when 1.3, 6.8, 13.4, and 58.2 seconds of speech data are used for speaker dependent parameter estimation.

Compensation Algorithm	Ave. Adaptation Data Dur. (sec)			
	1.3	6.8	13.4	58.2
Baseline	41.5	41.5	41.5	41.5
N	40.2	37.2	36.8	36.8
N+W	36.7	33.8	33.6	33.3
N+W+C	–	35.0	32.3	29.8
N+W+C+SAT	–	34.4	31.5	28.9

Table 1: WER obtained using unsupervised estimation of mean and variance normalization (N), frequency warping (W), and constrained model adaptation (C) parameters from varying amounts adaptation data.

There are several observations that can be made from Table 1. First, by comparing rows 1 and 2, it is clear that simply initializing mean and variance normalization estimates using the adaptation data (N) results in a significant decrease in WER across all adaptation data sets. Second, frequency warping (W) is also shown to provide significant reduction in WER with the most dramatic reduction occurring for the case where an average of only 1.3 seconds of adaptation data per speaker is used to estimate warping factors. Third, by observing rows 4 and 5 of Table 1, it is clear that constrained model adaptation (C) actually increases WER when the transformation matrix is estimated from less than 13.4 seconds of adaptation data. However, significant WER rate reductions were obtained as the adaptation data length was increased. It is important to note that the over-training problem observed here for adaptation algorithms resulting from insufficient adaptation data is well known. Future work will investigate the use of procedures that prevent over-training by interpolating counts estimated on a small adaptation set with those obtained from other sources of data [1].

4 CONCLUSIONS

Two developments associated with the implementation of robust mobile ASR services on hand-held devices have been presented. The first is an efficient framework for distributed mobile ASR based services. The DSEM server, presented in Section 2, was shown in Figure 2 to maintain acceptable response latencies with simultaneous ASR accesses from many hundreds of simulated mobile clients. The second is an efficient means for implementing robust acoustic compensation algorithms when there is little opportunity to influence the audio specifications of the device and little opportunity to sample all possible environments in HMM training. A set of acoustic compensation procedures, described in Section 3, were applied in an unsupervised user configuration scenario. These procedures, which include frequency warping based speaker normalization, constrained model adaptation, and off-line CMN and CVN, were shown in Table 1 to reduce word error rate by 30 percent.

ACKNOWLEDGMENTS

The authors would like to express their appreciation to Michiel Bacchiani for contributing his implementation of the CMA optimization algorithm.

REFERENCES

[1] A. Gunawardana and W. Byrne. Robust estimation for rapid speaker adaptation using discounted likelihood techniques. *Proc. Int. Conf. on Acoust., Speech, and Sig. Processing*, May 2000.

[2] M. J. F. Gales. Maximum likelihood linear transformations for HMM-based speech recognition. *Computer Speech and Language*, 12:75–98, 1998.

[3] M. Johnston, S. Bangalore, G. Vasireddy, A. Stent, P. Ehlen, M. Walker, S. Whittaker, and P. Maloor. MATCH: An architecture for multimodal dialog systems. *Proceedings of 40th Anniv. Mtg. of Assoc. for Computational Linguistics*, June 2002.

[4] L. Lee and R. C. Rose. A frequency warping approach to speaker normalization. *IEEE Trans on Speech and Audio Processing*, 6, January 1998.

[5] R. C. Rose, S. Parthasarathy, B. Gajic, A. E. Rosenberg, and S. Narayanan. On the implementation of ASR algorithms for hand–held wireless mobile devices. *Proc. Int. Conf. on Acoust., Speech, and Sig. Processing*, May 2001.

[6] O. Viikki. ASR in portable wireless devices. *Proc. IEEE ASRU Workshop*, December 2001.

[7] Matt Welsh, David E. Culler, and Eric A. Brewer. SEDA: An architecture for well-conditioned, scalable internet services. In *Symposium on Operating Systems Principles*, pages 230–243, 2001.