

A Tutorial on Distributed Speech Recognition for Wireless Mobile Devices

Dale Isaacs, A/Professor Daniel J. Mashao

Speech Technology and Research Group (STAR)
 Department of Electrical Engineering
 University of Cape Town, Rondebosch 7701, Cape Town, South Africa
 Tel:+27-21-650-4019, Fax:+27-21-650-3465
 dale@crg.ee.uct.ac.za, daniel.mashao@sita.co.za

Abstract—With the expansion in wireless communication technology and the introduction of powerful smart-phones, users are demanding systems which will allow for ubiquitous computing. A critical requirement is a simpler means of interacting with mobile devices. Instead of struggling with small keypads on smart-phones or a stylus on a PDA it would be much simpler if we could use a more natural and familiar medium of communication, speech. There are currently 3 architectures, Embedded Speech Recognition, Network Speech Recognition (NSR) and Distributed Speech Recognition (DSR), each with their own pros and cons, which aim to incorporate an Automatic Speech Recognition (ASR) system on mobile devices. DSR proposes to be the best solution due to its superior performance in the presence of transmission errors and noisy environments. The main aim of this paper is to give the reader a broad outline of the DSR architecture, but focuses mainly on the front-end system, which literature suggests is the most researched area of DSR. We present an outline of the current advanced front-end DSR standards in detail, investigating its architecture and possible permutations. We briefly touch on the back-end system of DSR and also look at issues relating to this technology.

Index Terms—Distributed Speech Recognition (DSR)

I. INTRODUCTION

Over the past 10 years there has been an exponential increase in the amount of mobile subscribers worldwide. In South Africa alone, market research has shown that by the end of 2007 the number of mobile device owners will be close to 30 million. In-Stat/MDR predicts that the worldwide wireless market will expand to more than 2.5 billion mobile subscribers by 2009. This together with the unprecedented development of the telecommunication industry over the last decade has brought about the need for ubiquitous access to a host of different information resources and services.

Despite the power of today's smart-phones and PDA's which allow us to perform tasks which previously were only available on a desktop/laptop computer, they are still limited in terms of size and input modalities. A simpler way of interacting with any device is via the most natural medium of communication, speech. The implications of communicating with any device via speech are immense. Taking just one simple example of the Short Message Service (SMS); instead of typing out a message

using small keypads or on an onscreen keyboard as in the case of most PDA's, just speaking a message to your device and having it automatically converted into text and ready for deployment. Other applications include dictation systems and speech information retrieval systems. This would bring a true meaning to the phrase, hands free communication.

Implementing an Automatic Speech Recognition (ASR) system onto any mobile terminal is not a trivial matter, due to its limitations in terms of physical size, processing power and memory capabilities (which will be discussed later in Section 2). Thus far, there have been three main architectures proposed for this type of application, an *Embedded Speech Recognition* system, a *Network Speech Recognition (NSR)* system and a *Distributive Speech Recognition (DSR)* system. In [1], the authors describe these architectures in more detail but conclude that DSR, with its superior performance in the presence of transmission errors and noisy environments will outperform both Embedded ASR systems and NSR systems due to its hybrid approach. In this paper we present a detailed overview of a DSR system, describing its architecture, the different techniques used on most systems as well as the issues concerned with it.

A typical DSR system is based on decoupling the front-end processing from the rest of the recognition mechanism using a client/server model over the data network [2]. The feature extraction and the speech recognition is distributed across the network and hence the name, Distributive Speech Recognition. This type of setup allows a terminal device (client) to only be responsible for feature extraction and speech coding part while the back-end server (central host) handles the decoding and computational extensive recognition part. Figure 1 shows an outline of a DSR system.

As shown in Figure 1, at the client, features are extracted (using a mel-cepstrum based feature extraction technique) from the input speech and then compressed and sent over the wireless channel to a large server where the features are decompressed and sent to a state-of-the-art Hidden Markov Model (HMM) based classifier. The classifier will return a recognition result back to the client.

The remainder of this paper is organized as follows, Section 2

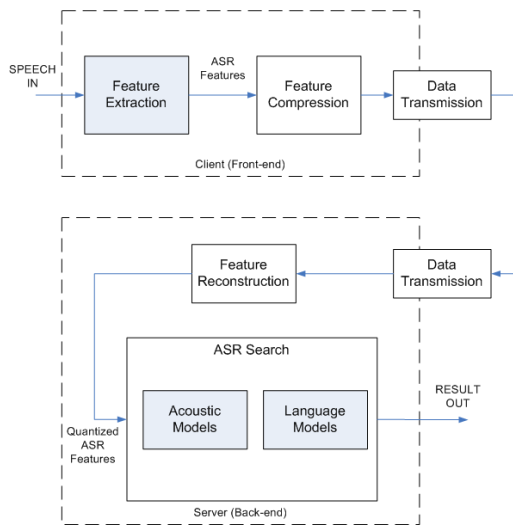


Figure 1. Client-Server based ASR system

discusses issues and benefits related to the implementation of ASR systems on mobile devices, Section 3 describes in more detail the front-end of a DSR system, Section 4 describes the back-end and finally, Section 5 gives a brief summary of the paper.

II. RELATED ISSUES AND BENEFITS OF DSR

When attempting to implement an ASR system onto any mobile device there will always be certain drawbacks which have to be taken into account due to the restrictions one has on the client side. Although DSR is a promising technology, it still has many issues which arise when implementing it in the real world but also has positive spin-offs which will be discussed in this section

A. Issues

The following is a list of challenges facing us when trying to implement the above mentioned type of system:

- physical size of the client device
- amount of processing power which the device is capable of obtaining
- amount of battery power which the system would require to operate efficiently
- amount of memory which the system needs to operate

Since we are focusing specifically on DSR and have our back-end running the recognition server most of the issues listed above are alleviated. However they would still need to be taken into account when implementing our feature extraction algorithm and speech coding on the client side. In [3], the authors performed an extensive study on the issues related to DSR and also explain that when using the mobile voice network, there is a degradation in performance due to low bit rate speech coding and channel transmission errors. In [3] it is also suggested that using an error-protected data channel will result in higher recognition performance.

B. Benefits

The main benefits of DSR are listed and then further explained below [4]:

- Improved recognition performance over wireless channels
- Ease of integration of combined speech and data applications
- Ubiquitous access with guaranteed recognition performance levels

The greatest benefit of DSR is the fact that the system is implemented on an error-protected data channel which minimizes the impact of speech codec and channel errors. This historically improves the performance of a recognition system over mobile speech channels. The use of DSR also enables multi-modal speech applications to operate over a single wireless data channel as opposed to having separate speech and data channels. DSR also offers the promise of a guaranteed level of recognition performance over every network.

III. DSR FRONT-END

A standardized Advanced Front-end (AFE) for DSR has been specified by the Aurora Working Group within the European Telecommunications Standards Institute (ETSI) for use on mobile phones and other communication devices which connect to speech recognition servers [5]. This has been done so that all front-end clients are identical regardless of the type of device the client is. This section describes this standard in more detail and was extracted from Aurora-ETSI Standards document [5]. A detailed diagram of the front-end is shown in Figure 3.

Some of the standards specified by ETSI are shown below [4]:

- Mel-Cepstrum feature set consisting of 12 cepstral coefficients logE and C0
- Data transmission rate of 4.8 kbps
- Low computational and memory requirements
- Low latency
- Robustness to transmission errors

A. DSR Mel-Cepstrum Front-end Standard

Figure 2, taken from [4] takes a closer look at the processing stages for a DSR front-end.

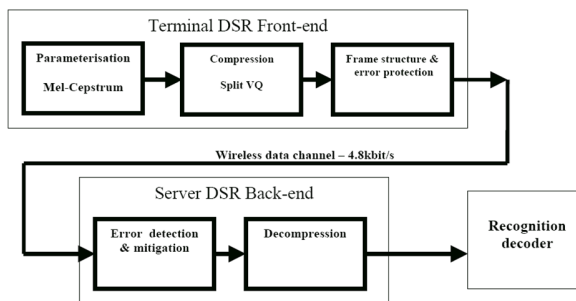


Figure 2. Detailed Block Diagram of DSR system

The following procedures are run [5]:

- Speech signal is sampled and parametrized using mel-cepstrum algorithm
- This generates 12 cepstral coefficients (see equation 1) along with C0 and logE (see equation 2)

$$C_i = \sum_{j=1}^{23} f_j \cdot \cos\left(\frac{\pi \cdot i}{23} (j - 0,5)\right), \quad 0 \leq i \leq 12 \quad (1)$$

$$\log E = \ln\left(\sum_{i=1}^N S_{of}(i)^2\right) \quad (2)$$

- Compressed to obtain lower data rate (4.8 kbps) for transmission
- Compressed parameters are formatted into defined bit-stream
- Then transmitted over wireless/wireline transmission link to a remote server
- Parameters are then checked for transmission errors
- Front-end parameters are decompressed to reconstruct the DSR Mel-Cepstrum features
- These are then passed to the recognition decoder sitting on central server.

B. Mel-Cepstrum Parametrization

Below is a block diagram showing the specification of the Mel-Cepstrum DSR Front-end standard submitted by Nokia.

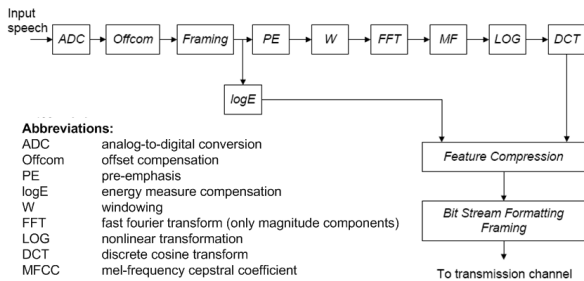


Figure 3. Block diagram of the ETSI front-end algorithm.

The feature vector, as mentioned above, consists of 12 cepstral coefficients (C1 - C12) together with C0 and logE (log energy) parameter, making up a total of 14 components. It is suggested in [4] that the reason for incorporating C0, was to support algorithms which would be requiring it at the back-end (such as noise adaption). Further details of the cepstral analysis are shown below [6].

- Signal offset compensation with notch filtering
- pre-emphasis with a factor of 0.97
- Application of Hamming window
- FFT based mel filterbank with 23 frequency bands in the range from 64Hz up to half of the sampling frequency

C. Feature Compression Algorithm

The standard for the compression algorithm was designed by Motorola and the specifications are discussed in this section. The compression algorithm was designed to take feature parameters for each short-time analysis frame of speech data (see equation 3) where m denotes the frame number plus C0 and logE (see equation 4)

$$c(m) = [c_1(m), c_2(m) \dots c_{12}(m)]^T \quad (3)$$

$$y(m) = \begin{bmatrix} c(m) \\ c_0(m) \\ \log[E(m)] \end{bmatrix} \quad (4)$$

A split vector quantization (VQ) algorithm is used to obtain final data rate of 4.8 kbps of speech. The closest VQ centroid if found using a weighted Euclidean distance to determine the index (see equations 5 and 6).

$$d_j^{i,i+1} = \begin{bmatrix} y_i(m) \\ y_{i+1}(m) \end{bmatrix} - q_j^{i,i+1} \quad (5)$$

$$idx^{i,i+1}(m) = \underset{0 \leq j \leq (N^{i,i+1} - 1)}{\arg \min} \left\{ \left(d_j^{i,i+1} \right)^T W^{i,i+1} \left(d_j^{i,i+1} \right) \right\}, \quad i = 0, 2, 4, \dots, 12 \quad (6)$$

A codebook of size 64 is used for each pair of cepstral coefficients from C1 - C12 and 256 vectors are used for C0 and logE (see Table I). A quantization scheme is used to code the relevant coefficients which result in 44 bits per speech frame. Error detection bits are added (4 bits of CRC for each pair of speech frames) to the compressed data and the compressed speech frames are grouped into multiframes for transmission and decoding (see Figure 4), this is done to counteract the presence of channel errors. For a more detailed description of this algorithm, see [5].

Codebook	Size ($N^{i,i+1}$)	Weight Matrix ($W^{i,i+1}$)	
		Element 1	Element 2
$Q^{0,1}$	64		C ₁ C ₂
$Q^{2,3}$	64		C ₃ C ₄
$Q^{4,5}$	64		C ₅ C ₆
$Q^{6,7}$	64		C ₇ C ₈
$Q^{8,9}$	64		C ₉ C ₁₀
$Q^{10,11}$	64		C ₁₁ C ₁₂
$Q^{12,13}$	256	Non-identity	C ₀ log[E]

Table I
SPLIT VECTOR QUANTIZATION FEATURE PAIRINGS [5]

Sync Sequence	Header Field	Frame Packet Stream
..-< 2 Octets ->..	...-< 4 Octets ->-< 138 Octets ->

<- 144 Octets Total ->

Figure 4. DSR Multi-frame Format

D. Possible permutations of AFE for DSR

The aforementioned components form the basis of the Aurora-ETSI standard for a front-end feature extraction and compression algorithm. It is important to note that this is not the only method being used but merely a guideline.

There have been researchers who propose to improve this technique, as in [7] where the authors use a novel approach to improve the computational efficiency. They use a structure of two-stage mel-warped Wiener filtering algorithm which is the main part for AFE. Using this approach discards the convolution operations in time-domain and the calculation of the power spectrum thereby saving on a large number of computations.

In [8], a proposal was made to reduce the recognition complexity by compressing the extracted features using scalable encoding techniques which provides a multi-resolution bitstream together with a scalable recognition system. In this paper it was shown that using speech coders optimized for recognition rather than perceptual distortion provides a better recognition rate performance.

An investigation into the use of Gaussian Mixture Models (GMMs) for the coding of Mel frequency-warped cepstral coefficient (MFCC) features in DSR is shown in [9]. In this paper the authors compare vector quantizers to a GMM-based block quantiser which has relatively low computational and memory requirements. Their studies show an improvement in recognition performance due to the simple computations and bit-rate scalability.

In [10], another interesting approach is taken whereby the authors present and integration of hybrid acoustic models using tied-posteriors in the distributive environment. Their results show that a hybrid technique is able to outperform standard continuous systems. Their approach proves to be useful since changes can be made to the client without affecting the server and vice versa.

IV. DSR BACK-END

The back-end system of DSR system is where the recognition takes place and this is usually done on a high powered desktop machine which can take advantage of its processing power and memory capacity which is the main concept giving DSR the edge over other attempts to implement ASR systems on mobile devices.

A traditional back-end server in a DSR system would usually contain a feature reconstruction part and also a recognizer as shown in Figure 5. The feature reconstruction model would be responsible for decoding the feature vectors coming from the front-end and also checking it for transmission errors. The reconstructed cepstral features would be sent to the recognizer which then present its results.

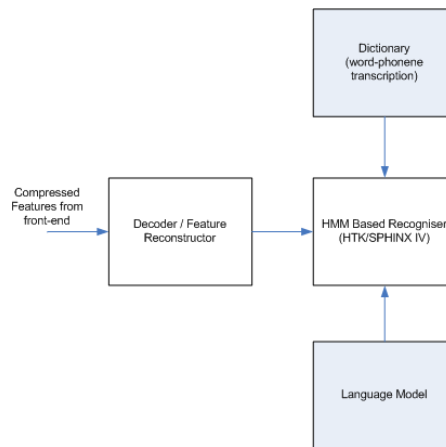


Figure 5. Traditional Back-end block diagram

The two most popular state-of-the-art speech recognizers are the Hidden Markov Model Toolkit (HTK) and SPHINX IV. Just like the front-end it is only a template of components the server should contain. An IBM research report [11], discusses the server-side speech reconstruction in more detail but was out of the scope of this report.

There have also been proposals in [12] to reduce the recognition complexity at the server-side. Their research investigates a scalable recognition system to perform this task to reduce both the computational load and the bandwidth requirement at the server by using a low complexity pre-processor to eliminate unlikely classes.

V. SUMMARY

With today's technology growing at such a rapid pace, mobile users are making use of their devices more frequently and require access to information at the touch of a button. New applications are being launched everyday but consumers are demanding better interfaces for their mobile devices. Using such a natural medium as speech, will mean true hands free communication. In this paper we outlined and reviewed a promising ASR technique for mobile computing, DSR, looking at its current standards.

We looked in detail at the advanced front-end algorithm set by the authors of [5][6] and also reviewed other alternatives to the standards which aim to improve the overall performance on this system. This paper also showed the structure of the back-end and also issues relating to DSR.

VI. ACKNOWLEDGMENTS

I would like to thank the National Research Foundation (NRF) and the Telkom Centre of Excellence (CoE) for their continued financial support thereby making this research possible.

REFERENCES

- [1] D. Zaykovskiy, "Survey of the speech recognition techniques for mobile devices," Department of Information Technology, University of Ulm, Germany, 2006.
- [2] M. Perakakis, "Distributed speech recognition," <http://www.telecom.tuc.gr/perak/speech>, 2001.
- [3] P. Manolis, "Distributed speech recognition issues," tech. rep., Department of Electronics and Computer Engineering, Technical University of Crete, June 2001.
- [4] D. Pearce, "Enabling new speech driven services for mobile devices: An overview of the etsi standards activities for distributed speech recognition front-ends," in *AVIOS: The Speech Applications Conference*, (Motorola Limited, Jays Close, Viabes Industrial Estate, Basingstoke, HANTS, RG22 4PD, United Kingdom), Motorola Labs and Chairman ETSI STQ-Aurora DSR Working Group, May 22 - 24 2000.
- [5] ETSI, "Speech processing, transmission and quality aspects (stq); distributed speech recognition; front-end feature extraction algorithm; compression algorithms," Document ETSI ES 201 108 V1.1.2 (2000 - 2004), European Telecommunications Standards Institute, <http://www.etsi.org>, 2000.
- [6] D. Pearce and H. Hirsch, "The aurora experimental framework for the performance evaluation of speech recognition systems under noisy conditions," in *ICSLP 2000(6th Int Conf on Spoken Lang Processing)*, (Beijing, China), October 2000.
- [7] J. Li, B. Liu, R. Wang, and L. Dai, eds., *A Complexity Reduction of ETSI Advanced Front-end for DSR*, vol. 1 of *ICASSP*, iFlytek Speech Lab, University of Science and Technology, China, 17-21 May 2004.
- [8] N. Srinivasamurthy, A. Ortega, and S. Narayanan, "Efficient scalable encoding for distributed speech recognition," 2003.
- [9] S. So and K. Paliwal, eds., *Scalable distributed speech recognition using Gaussian mixture model-based block quantization*, vol. 48 of *Speech Communication*, (Brisbane QLD 4111, Australia), School of Microelectronic Engineering, Griffith University, 2006.
- [10] J. Stadermann and G. Rigoll, eds., *Hybrid NN/HMM acoustic modeling techniques for distributed speech recognition*, vol. 48 of *Speech Communication*, (Munich, Germany), Technische Universität München, Institute for human-machine communication, August 2006.
- [11] T. Ramabadran, A. Sorin, M. McLaughlin, D. Chazan, D. Pearce, and R. Hoory, "The etsi extended distributive speech recognition (dsr) standards: Server-side speech reconstruction," Research Report H-0200, IBM, October 22, 2003 2003.
- [12] N. Srinivasamurthy, A. Ortega, and S. Narayanan, "Efficient scalable speech compression for scalable speech recognition," in *IEEE International Conference on Multimedia and Expo*, Integrated Media Systems Centre, Dept of EE-Systems, University of Southern California, Los Angeles, CA 90089-2654, 2000.

Dale B. Isaacs is currently pursuing a MSc (Eng) degree in Electrical Engineering at the University of Cape Town and is a student of the STAR group.

A/Professor Daniel J. Mashao is the supervisor of the above mentioned author at the University of Cape Town (UCT) and Chief Technical Officer (CTO) of the State Information Technology Agency (SITA).