Robust Speech Recognition in Client-Server Scenarios

Richard Rose\(^1\) and Hong Kook Kim\(^2\)

\(^1\)Dept. of Electrical and Computer Engineering
McGill University, Montreal, QC Canada
rose@ece.mcgill.ca

\(^2\)Dept. of Information and Communications
Gwangju Inst. of Science and Technology, Korea
hongkook@gist.ac.kr

Abstract

This paper addresses issues that are specific to the implementation of automatic speech recognition (ASR) applications and services in client-server scenarios. It is assumed in all of these scenarios that functionality in a human-machine dialog system is distributed between mobile client devices and network based multi-user media and application servers. It is argued that, while there has already been a great deal of research addressing issues relating to the communications channels associated with these scenarios, there are many additional problems that have received relatively little attention. These include issues of how environmental and speaker robustness algorithms are implemented in mobile domains and how multiple ASR channels can be implemented more efficiently in multi-user deployments. Preliminary results are summarized showing the effect of user specific unsupervised adaptation and normalization algorithms on ASR performance in mobile domains. Results are also presented demonstrating the efficiencies that are obtainable from using intelligent algorithms for assigning ASR decoders to computation servers in multi-user deployments.

1. Introduction

There are many issues associated with the implementation of robust ASR when speech enabled services are delivered in a client-server configuration. There are also problems that are specific to maintaining high quality of service for ASR under peak load conditions in multi-user network deployments. These issues are dictated by the communications channels connecting client and server, the class of speech-enabled services and devices that are supported, and by the platforms that they run on. The majority of the research in this area has focused on the effect of communications channels on ASR performance. However, this paper argues that there are opportunities for making a considerable impact on speaker and environment robustness and overall system efficiency in the context of client-server system implementations. Section 2 of the paper summarizes previous work performed on feature extraction, channel decoding, and acoustic modeling to compensate for channel distortions in wireless and VoIP channels. The remainder of the paper attempts to motivate the need for research in other areas related to client-server scenarios.

Examples of services with a unique set of robustness issues are ASR services on personalized mobile devices. These are often implemented in a client-server mode and represent application domains that can often be characterized by continuously varying, sometimes low signal-to-noise ratio acoustic environments. However, the use of personalized portable devices also presents the opportunity for an adaptation/normalization paradigm where parameters can be continually re-estimated to better characterize a particular user, environment, or device. An investigation of algorithms and architectures for implementing these paradigms is presented in Section 3.

Examples of platforms in client-server scenarios where there exist very specific efficiency issues are multi-user deployments that make use of many low cost commodity servers. Rather than simply improving the efficiency of individual ASR engines, the important issue is to increase the efficiency of the multi-user configuration at peak loads. This is investigated in Section 4 through the use of intelligent strategies for assigning ASR decoders to application servers.

2. Robustness Issues for Mobile Domains

In client-server based scenarios, speech may be transmitted over a variety of wireless and wireline communications channels and voice over Internet protocol (VoIP) networks. There has been a great deal of research addressing issues of acoustic feature extraction and channel robustness for ASR in these communications frameworks \([1,2,3,4,5,6,7]\). This section provides a brief summary of some of this work. It is also suggested in Section 2.3 that other sources of variability can contribute to ASR performance degradation, especially in mobile applications. This serves as motivation for the discussion in Section 3 on normalization and adaptation algorithms for compensating for multiple sources of variability.

2.1. Feature Extraction Scenarios

There have been several investigations comparing the ability of different feature analysis scenarios to obtain high performance network-based ASR over wireless telephone networks \([3,4,5,6]\). The ETSI distributed speech recognition (DSR) effort has standardized feature analysis and compression algorithms that run on the client handset \([3]\). In this scenario, the coded features are transmitted over a protected data channel to mitigate the effects of degradation in voice quality when channel carrier-to-interference ratio is low. Another scenario involves performing feature analysis in the network by extracting ASR features directly from the received voice channel bit stream \([5]\).

A last scenario has been evaluated for ASR which does not involve additional client based or network based processing \([4,6]\). Instead, it involves the use of the adaptive multi-rate (AMR) speech codec that has been selected as the default speech codec for use in Wideband Code Division Multiple Access (WCDMA) networks. Studies have shown that the
ability of the AMR codec to trade-off source coding bit-rate over a range from 4.75 to 12.2 kbit/s with channel coding bit allocation results in negligible change to ASR accuracy for carrier to interferer ratios as low as 4 dB [4].

2.2. Robustness with Respect to Channel Distortions

There have also been a variety of techniques that have been investigated for making ASR more robust with respect to the distortions induced by Gaussian and Rayleigh fading channels associated wireless communications networks [2,7]. The most interesting of these techniques involves a combination of confidence measures derived from the channel decoder with the likelihood computation performed in the Viterbi search algorithm used in the ASR decoder. In these techniques, confidence measures are computed from the a posteriori probability that received feature vectors are correctly decoded. These confidence measures are then used to weight or censor the local Gaussian likelihood computations used in the Viterbi algorithm [2,7].

These approaches are similar in some ways to the “missing feature theory” approach to robust ASR where noise corrupted feature vector components are labeled and removed from the likelihood computation [7]. However, the ability of the channel decoder to identify missing features in this application is far more effective than the existing techniques for labeling feature vectors corrupted by noisy acoustic environments. Similar techniques have been investigated for the packet loss scenarios associated with packet-based transmission over VoIP networks [1].

2.3. Importance of Acoustic Environment

It is well known that distortions introduced by both the acoustic environment and the communications channel can impact ASR performance. Studies based on empirical data collected in multiple cellular telephone domains have demonstrated that the effects of environmental noise can often dominate the observed degradations in ASR word error rate (WER) [8]. Decreases in WER of 50% have been measured over wireless communication channels in noisy automobile environments compared to quiet office environments. On the other hand, a WER decrease of only 10% was observed in wireless channels compared to wire line channels when speech was collected in a quiet office environment. This agrees with similar findings suggesting that, except for extremely degraded communications channels, the impact of channel specific variability can be secondary with respect to environmental conditions in mobile ASR applications.

3. Adaptation / Normalization Algorithms

This section describes an architecture that is used for implementing robust acoustic normalization / adaptation algorithms in a client-server framework [9]. The adaptation paradigm assumes that there is continuous, incremental updating of adaptation parameters in an unsupervised mode. The architecture is implemented in the context of the distributed speech enabled middleware (DSEM) framework [9]. The section has three parts. First, the DSEM will be described. Second, the adaptation/normalization architecture will be presented. Finally, ASR results for this adaptation paradigm will be briefly summarized.

3.1. Distributed Speech Enabled Middleware

The DSEM framework borrows from work done in obtaining efficient communications infrastructures for minimizing overheads associated with accessing internet servers [9]. Traditional server implementations assign a separate thread or process per client which works well until the number of clients becomes very large. At this point overheads associated with context switching and contention for resources starts to limit the overall throughput. This problem is exacerbated by the large IO requirements associated with ASR. The DSEM deals with these problems by using an event-driven, non-blocking IO model with only a single thread to manage all concurrently connected clients.

The block diagram in Figure 1 describes the DSEM framework in terms of its functional components. The figure illustrates how the DSEM provides the communication path between the mobile clients connected through the wireless network and various application servers including a cache of ASR decoder instances, HTTP servers, and database servers. It can be described by using an example client-server interaction where a voice query is made from the mobile user to the server, the decoded ASR result is used to issue an HTTP query, and a result is returned and displayed on the mobile client.

The dispatcher is responsible for detecting and routing all of the system’s IO events. So when the client initiates a request and streams audio to its session, the dispatcher detects the request and notifies its session object about the audio stream coming from the client. The session object then invokes an appropriate handler which in this case performs acoustic feature analysis, invokes various acoustic feature normalization and adaptation algorithms, and invokes a decoder instance from the cache of decoder proxies maintained by the DSEM framework. When the ASR result is returned, the handler uses it to perform a query to the HTTP server by invoking an HTTP object. When that query result is received, the handler can then return a message to be displayed on the client.

Figure 1: Distributed speech enabled middleware (DSEM)

3.2. Adaptation/Normalization Architecture

Figure 2 illustrates how unsupervised adaptation/normalization algorithms and the notion of a configuration server are implemented in the DSEM framework. The DSEM framework is again shown as providing an interface between the client devices connected over a wireless network and application servers. In addition to the speech recognition servers, we have also depicted a set of servers that
are responsible for performing these parameter re-estimation procedures which are referred to here as configuration servers. A mechanism is also depicted for logging speech data and ASR results in the DSEM that are used to drive the estimation algorithms in the configuration server. The configuration servers produce the user specific adaptation parameters that are used in feature analysis for the target user.

At recognition time, audio is streamed from the mobile client to the handler associated with a particular application. Acoustic features are extracted, feature space adaptation/normalization procedures are applied, and the acoustic features are passed along to the ASR decoder. At infrequent intervals, accumulated logging information for a particular user is passed to a configuration server and the associated adaptation parameters are re-estimated.

3.3. Adaptation/Normalization performance

A study was performed to demonstrate the performance of several adaptation/normalization algorithms that were implemented according to the paradigm described in Section 3.2 [9]. Several different techniques were applied in the feature space domain in order to reconfigure a speaker independent, task independent ASR system in an unsupervised, incremental update mode. The first technique, frequency warping based speaker normalization (WARP), estimates a single parameter linear frequency warping coefficient to minimize the mismatch between the input utterance and the speaker independent hidden Markov model (HMM) [10]. The second technique, cepstrum mean normalization (CMS), removes an estimate of the mean from the observation vectors to minimize the mismatch of long term statistics between test and training conditions. The third technique, constrained model space adaptation (CMA), estimates a matrix transformation to the feature vectors which is estimated to maximize the likelihood of the adaptation data with respect to the HMM model [11]. CMA was performed in association with speaker adaptive training (SAT).

These techniques were evaluated on a 3000 name directory retrieval task for speakers interacting with a multimodal user interface on a Compaq Ipaq in an office environment using the standard device mounted microphone [9]. The results are summarized in Figure 3 where the word error rate is plotted with respect to the number of seconds of adaptation data that were used for estimating parameters. There are several observations that can be made from these results. First, it is clear that the combination (CMS+WARP) achieves a significant WER reduction with as little as two seconds of speech. Second, when these procedures are combined with CMA (CMS+WARP+CMA+SAT), it is clear that substantial additional WER reduction is achieved, but only when there is greater than approximately ten seconds of adaptation speech data.

Figure 3: WER plotted versus adaptation speech data for frequency warping based speaker normalization (CMS+WARP) and for the combination of warping with CMA and SAT (CMS+WARP+CMA+SAT).

4. Efficient Use of ASR Resources

When ASR is deployed in a multi-user client-server scenario, it is important that the system can support a large user population while simultaneously minimizing degradation in quality of service under peak load conditions. The high degree of variability in processing effort that exists for ASR in human-machine dialog scenarios makes this a challenging problem. One source of variability results from the infrequent occurrence of user utterances as a fraction of the total length of a user-machine dialog. If ASR decoders are allocated to application servers for an entire call, this variability is unavoidable. Other sources of variability include the high variance in processing load over time in an utterance and the high variance in processing load that exists between different ASR tasks. It is the first of these sources of variability that is the most obvious and the easiest to address.

Most multi-user deployments rely on a call level allocation (CLA) strategy for assigning ASR decoders to application servers. This corresponds to the assignment of a decoder from the time the call is connected to the time it terminates. Figure 4 demonstrates the potential benefits that can be obtained by invoking a simple utterance level allocation (ULA) strategy for assigning ASR decoders to servers [12]. The figure shows a media server which directs calls to a resource manager. The resource manager detects the existence of utterances in a call, performs feature analysis, and then allocates an ASR decoder to one of two ASR servers. Each horizontal line extending from the media server in Figure 4 corresponds to the time-line of a call with individual utterances depicted by rectangles. The example shows a situation where, if a call level allocation strategy were used, many overlapping utterances could result in an individual server being overloaded. In the example, the resource manager tracks all utterance activity and dispatches individual utterances to balance the load on each server.
adapt a speaker independent HMM system to individual users of mobile handheld devices. When approximately one minute of adaptation utterances were used to estimate adaptation parameters in a large vocabulary name recognition task, a 31% reduction in word error rate was obtained. An intelligent scheme for allocating ASR decoders to application servers in multi-user client-server deployments was shown to decrease average response latencies by over a factor of two. Further work should study how to address the architectural impact of these procedures and how they can be realized in a variety of infrastructures.

References


5. Conclusions

This paper has demonstrated two procedures for improving ASR robustness and efficiency specifically in multi-user client-server scenarios. A combination of techniques for unsupervised acoustic adaptation and normalization were applied in the feature domain to

The results of an empirical study comparing CLA and ULA strategies are given by the plots in Figure 5 [12]. Four hundred calls of natural language queries were presented simultaneously to a multi-user system. The multi-user system corresponded to a deployment of ten Linux servers, where each server was running at 1 GHz and able to process two simultaneous utterances without overload. On average, speech was active for 35 percent of the total call duration. The figure of merit used to evaluate the performance of the allocation strategies was the latency that would be observed by a user from the end of a spoken utterance to the time when a recognition result is returned by the ASR decoder.

Figure 5 displays the percentage of utterances for the ULA and CLA strategies that could be processed within a maximum response latency of 0.5 to 3.0 seconds as displayed along the horizontal axis. It is clear from the figure that the ULA strategy is able to handle twice as many calls with sub-second latencies than the CLA strategy. This is important because while there has been a great deal of effort applied to increasing the efficiency of individual ASR engines, there has been little effort applied to increasing overall efficiency at peak loads in multi-user scenarios.

Figure 5: Performance of CLA and ULA strategies