Robust Speech Recognition in Ubiquitous Networking and Context-Aware Computing

Zheng-Hua Tan, Paul Dalsgaard, Børge Lindberg, Haitian Xu

Centre for TeleInFrastructure (CTIF), Speech and Multimedia Communication (SMC)
Aalborg University, Denmark
{zt, pd, bli, hx}@kom.aau.dk

Abstract

The introduction of ubiquitous computing and networking has fostered automatic speech recognition (ASR) systems of a distributed nature. The major challenge in deploying ubiquitous ASR is that the operating environments may change rapidly leaving the ASR system very vulnerable. This paper deals with the concept of making ASR systems context-aware with the aim of improving robustness against varying conditions such as dynamic network constraints and environmental noise. To fully benefit from a variety of networks with different characteristics, a number of distributed speech recognition (DSR) schemes are presented each of which is applicable to a specific network context. To increase ASR system robustness in varying environmental noise context, a multiple-model framework for noise-robust ASR is presented where multiple HMM model sets are trained, one for each noise type and each specific signal-to-noise ratio (SNR) that characterise the noise context. Experimental results show that the performance of ASR is largely improved by exploiting the context information.

1. Introduction

The trend in recent years computing is that networking is becoming pervasive and devices are shrinking in size and becoming common in use, which together pave the way to a ubiquitous computing environment. This is both an opportunity and a challenge for the development of services of today. The opportunity is that services and systems can have higher complexity, consume more computing resources, and access to more information than ever before. The challenge, however, is that the operating environments in ubiquitous computing generally change rapidly making a static system unsuitable. Therefore there is a high demand for systems to adapt to the ever changing environment – the context. Context-awareness is forecasted to play a paramount role in the success of future services.

This transfer in computing will have a significant impact on the present ASR research, demanding also for a paradigm shift to take place. On the one hand, the advent of ubiquitous networking has promoted the development of ASR systems of a distributed nature which again facilitate ubiquitous ASR. The benefits offered by networked solutions such as DSR have been widely accepted. On the other hand, it is a challenge to deploy ubiquitous ASR in the varying contexts.

More generally, the deployment of ASR technology of today is restricted because of the variability in environmental noise, in channel induced noise and in speaker-related variations. In context-aware systems, however, some of these robustness problems can potentially be reduced. This may for example include improving system robustness by regularly monitoring and utilising environmental information such as noise, and eliminating speaker-to-speaker variability of speech by introducing the concept of a personalised recogniser, as envisioned by Furui [1].

At least three categories of context are foreseen for consideration: 1) the computing context such as network and terminal capabilities; 2) the user context such as the user's profile and location; 3) the physical context such as noise characteristics [2].

This paper reviews the concept of context-awareness as incorporated into ASR systems. Then two aspects in particular are introduced. The first is on network awareness where a number of DSR schemes are presented each of which is applicable to one specific network context. The second is on environmental noise awareness in which an ASR system based on noise type- and SNR classification is introduced.

2. Context and context-awareness

Though context is lexically defined as "the circumstances in which an event occurs" it gives rise to a number of different interpretations within ubiquitous computing. One way of defining context is to list examples often encountered. For example, Schilit et al. define context as the dynamically changing environment including computing environment, user environment and physical environment [2]. This definition is intuitive and instructive but is still insufficient in its capability to generalise. Instead, Dey and Abowd define context as "any information that can be used to characterise the situation of an entity, where an entity can be a person, a place, or a physical or computational object" [3].

The use of context is becoming increasingly important yet is still a challenging problem. A system is context-aware if it is enabled to collect, understand and utilise context information e.g. by adapting its behaviour to the current context. Adaptation is the key element of context-awareness.

As context is information that users do not explicitly provide, it is critical to automatically collect context knowledge [4] and ubiquitous networking significantly facilitates this collection. A noticeable development in this area is the introduction of sensor networks which capture the information of surroundings and enable services to act upon the revealed information [5].

3. Context-aware ASR systems

In traditional computer systems an input-output architecture is adopted where only *explicit* input and output are considered. Lieberman and Selker extend the architecture by including context as an - *implicit* - input and output to and from the system [6]. In ASR systems, the input and output are the speech signal and the recognised words, respectively. In making this system context-aware, a number of additional sources such as speaker-related information, acoustic environments, computing resources and the history need be taken into account as context knowledge, as shown in Fig. 1.

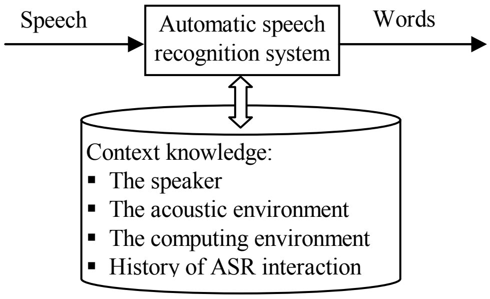
Speech
Automatic speech recognition system
Words
Context knowledge:
▪ The speaker
▪ The acoustic environment
▪ The computing environment
▪ History of ASR interaction

Figure 1: Architecture of a context-aware ASR system.

Robustness against the highly varying contexts has been the primary challenges in ASR technology over the last decades. The variations are extremely complex both in dimension and in range, making universal methodologies unrealistic so a paradigm shift is needed with the aim of designing systems sensitive to the contexts. An interesting work along this route is the prototype system developed by Rose et al. [7], where both acoustic and language models can be adapted to the device, user, and acoustic environment on the basis of the continually updated "configuration server".

In the following three categories of contexts are discussed and investigated in terms of their relevance to ASR systems.

3.1. Computing context

The computing context is mainly concerned with network connectivity, communication bandwidth and cost, device capacity and so on. Broadly speaking it encompasses the terminal-context and the network-context.

In the years to come it is expected that a growing variety of communication environments will emerge, and as a consequence specific research efforts have to focus on network context management with the aim of collecting, maintaining and disseminating context information [8]. Access to such context information is of importance for the continued development of e.g. DSR.

3.2. User context

In the domain of user context, personalisation is a key concept and is receiving increased emphasis. Personalisation of ASR can be reached by a number of ways such as training and adapting both acoustic models and language models for a specific user. Furthermore, the profile of the user, her or his devices and even service tasks can be stored in a centralised "personal ASR" server.

Instead of expanding speech databases with an increasing number of speakers, Shi and Chang use massive amounts of speaker-specific training data recorded in one's daily life with the aim of training the personalised acoustic models. This strategy has shown a substantial improvement in ASR performance as compared to speaker-independent system with speaker adaptation applied [9].

3.3. Physical context

Environmental noise is a key element of physical context particularly for ASR as the strong variability of acoustic noise may dramatically degrade the ASR performance. To collect noise information, a number of specially designed sensors may be either deployed in mobile devices or embedded in the environment as part of sensor networks that may provide contextual information to devices on the basis of their locations [2]. To exploit the noise context, Akbacak and Hansen introduce an environmental sniffing framework to improve ASR robustness [10].

4. Network and acoustic variations to ASR

Aimed at optimal performance of ASR over mobile and IP networks, an important research topic within ASR has been to focus on the issue of DSR [13]-[15], [18]. In the client-server architecture, the DSR system splits ASR processing into the client based front-end feature extraction and the server based back-end recognition, where data transmission between the two parts may take place via heterogeneous networks. However, the transmission of data across networks presents a number of challenges to speech technology, for example bandwidth limitations and transmission errors.

Additional to the network degradations, the speech signal is also corrupted by both transducer distortion and additive noise. Fig. 2 illustrates the architectural model including the degradations caused by both transmission errors and acoustic noise. How to handle these degradations in the context-aware framework is the focus of the following two sections.

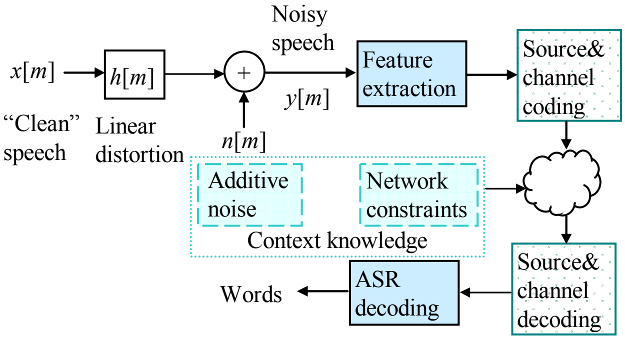

Figure 2: Model of ASR degradations.

5. Network awareness

As the range of communication environments becomes larger and larger, it is meaningful for networked systems – like DSR – to be responsive to the network context. Given the availability of context-aware networks in which relevant context information is established centrally for services, this section introduces a set of DSR schemes each of which matches a specific network context characterised by bandwidth, delay and type of networks .

5.1. DSR schemes

5.1.1. Half frame-rate

Both in low bandwidth networks and in high traffic situations, there is a need to have a bit-rate as low as possible (as in speech coding) while still maintaining acceptable recognition performance. Motivated by the redundancies observed in full frame-rate (FFR) features caused by both the overlapping in the feature extraction process and the speech production process itself, a half frame-rate (HFR) front-end is presented for DSR. At the client-side, the HFR is implemented simply by using the double frame shifting compared to FFR and therefore half the bit rate is achieved. At the server-side, each HFR feature vector is repeated once to construct an estimate of the FFR features and thus no modifications are needed in the recognition back-end. It is experimentally justified on small and medium vocabulary tasks that the performance attained by HFR is almost equal to FFR; however, repetition of each HFR feature vector is critical for the HFR front-end to maintain the performance [11]. In addition to its low bit-rate, the HFR front-end only requires half the computational cost of the FFR, which is a significant reduction for resource-limited hand-held devices.

5.1.2. Multiple description coding (MDC)

In the HFR front-end the extracted feature vectors simply correspond to the odd-numbered feature vectors of the FFR feature vectors. Together with the even-numbered feature vectors, two descriptions of the speech signal are generated and each of them can independently be transmitted, resulting in a MDC coding scheme. Unlike most conventional coders, MDC encodes a source into two or more sub-streams (descriptions) that each can be delivered on separate channels with the aim of exploiting channel diversity and thus improving robustness against transmission errors [12].

5.1.3. Interleaving

Interleaving techniques have been broadly applied in communication systems to counteract the effect of burst errors, but at the cost of transmission delay. In interleaving, the ordering of a sequence of code symbols is rearranged with the aim of spreading the burst errors over multiple code words for efficient error concealment (EC). At the server, the counterpart de-interleaving restores the rearranged sequence to its original order. Interleaving has shown good performance for DSR [13]. In this paper, interleaving is implemented by re-ordering the odd-numbered and the even-numbered feature vectors.

5.1.4. Sub-vector error concealment

Conventional EC algorithms share the common characteristic of conducting EC at the vector level. In circuit-switched networks, however, transmission errors mainly occur at the bit level, which results in the fact that within erroneous vectors a substantial number of sub-vectors are often error-free. Evidently, the vector-level strategy fails to exploit error-free fractions left within erroneous vectors. In sub-vector based EC [14], the detected erroneous vectors are submitted to a further analysis where each sub-vector is analysed individually. This is conducted on the basis of a data consistency test applied to each erroneous vector with the aim of identifying inconsistent sub-vectors and resulting in a consistency matrix. On the basis of the consistency matrix each inconsistent sub-vector is replaced by its nearest neighbouring consistent sub-vector whereas consistent sub-vectors are kept untouched. This technique has shown to be suitable for DSR encoded by split vector quantization (SVQ) and for circuit-switched network transmission.

5.2. Experimental settings

The Aurora 2 database [15] has been selected. The database is the TI digit database artificially distorted by adding noise and using a simulated channel distortion. Whole-word models are trained for all digits using the HTK recogniser. Each of the digit whole word models has 16 HMM states with three Gaussian mixtures per state. The silence model has three HMM states with six Gaussian mixtures per state. A one-state short pause model is tied to the second state of the silence model. To simulate transmission errors, the widely used GSM EP3 (error pattern) is chosen for this evaluation [14].

5.3. Experimental results and discussion

The HFR is implemented by using a frame shift of 20 ms. In applying MDC, the two description encodings are transmitted over two uncorrelated channels which both are simulated by EP3. Two interleaving schemes are applied: Interleaving12 in which a sequence of 12 vectors is grouped into one block and Interleaving24 where a sequence of 24 vectors is grouped. Interleaving is implemented simply by reading odd-numbered features first and even-numbered features second from the blocks. As a result, Interleaving12 and Interleaving24 have 50 ms (or 5 vectors) and 110ms maximum delay, respectively. Repetition (used in Aurora baseline), linear interpolation and sub-vector EC are also evaluated.

Table 1 shows the performance resulting from applying the different DSR schemes on the clean data from Test Set A. It is observed that the performance of the MDC scheme approaches that of the error-free channel. The restriction of deploying MDC is the requirement of available independent multiple channels. The interleaving schemes achieve good performance, however at the expense of an added delay. Sub-vector EC gives good performance with the requirement of errors occurring at the bit level. HFR performs better than the Aurora baseline with only half the bit rate.

Future networks will be dynamic and heterogeneous in nature and each type of network has its own characteristics. To adapt to the changing network environment, different DSR schemes are needed and the DSR scheme should be chosen on the basis of the network context.

Table 1: Averaged word error rate (WER) for the DSR schemes for clean data from Test Set A imposed by EP3

	WER (%)		WER (%)
Repetition (baseline)	6.70	Interleaving12	2.43
Interpolation	7.35	Interleaving24	1.74
Half frame rate	4.56	MDC	1.04
Sub-vector	2.65	**Error-free**	0.95

6. Environmental noise awareness

6.1. A multiple-model framework

Dynamically changing noise conditions (e.g. type and SNR) and mismatch between training and test environments often cause significant degradations in ASR performance. To handle this, Lippmann et al. propose a multi-style training (MTR) where acoustic models are trained using a speech corpus corrupted by noises expected to appear in the user environment. An alternative way of handling such mismatches is to segregate the noise into a number of noise types and for each type to train a set of HMM models [10]. A further improvement to this approach is to expand the noise characteristics by dividing each noise type according to its SNR, resulting in a SNR and Noise Classification based Multiple-Model Framework (SNC-MMF) [16]. By introducing such sub-division, environmental noise context can now be parameterised on the basis of noise type and SNR.

In SNC-MMF, different HMM model sets are built for each combination of SNR and noise type. The efficiency of the ASR decoding is maintained by selecting only one model set according to the estimation of noise type and SNR. It has been experimentally justified that with only three model sets for each known noise type, significant improvement is obtained for the known noise types as compared to the MTR method while the performance for the unknown noise type is lower, due to the training-test mismatch. However, the well known Jacobian (JAC) adaptation method may be used to reduce the mismatch in the model domain [17]. Since the SNR mismatch is alleviated in the SNC-MMF, a modification of the JAC has shown advantageous and introduced by using zero noise-level difference Jacobian (Z-JAC). Here only the difference in noise-type is handled while the noise level is kept untouched and the modification involves setting the difference of the noise energy component to zero.

Since MTR models are generally robust against unknown noise due to its enclosure of information from a variety of noise environments, model interpolation is presented to interpolate the selected SNR and noise specific models with the MTR ones.

6.2. Experimental results and discussion

Evaluation is also conducted on the Aurora 2 database as described in Section 5.2. In the three test sets, the four noise types in Set A are treated as the known noise type, the four in Set B are the unknown noise type and Set C includes one known and one unknown noise type in addition to convolutional noise. For each of the four known noise types, noise data and clean speech training data are artificially merged with SNR values close to 5dB, 10dB and 20dB, generating data for the training of three HMM model sets. A simple voice activity detection based SNR estimator and a cepstral GMM based noise classifier are used. Table 2 shows that the basic SNC-MMF modelling gives significant improvement over MTR modelling for Set A but lower performance for Set B. The combination of the model interpolation and Z-JAC in SNC-MMF gives the best performance showing a relative WER reduction of 24.5%.

Table 2: WER (%) for different test sets and relative improvement (%) compared to MTR

	Set A	Set B	Set C	Average	Improv.
MTR	12.18	13.73	16.22	13.61	--
SNC-MMF	8.15	16.99	13.56	12.77	6.2
Interp. +Z-JAC	8.05	11.41	12.45	10.28	24.5

7. Conclusions

The paper reviews research in context-awareness applied in the ASR domain. Attention is particularly paid to network awareness and environmental noise awareness. On network awareness, a number of DSR schemes are presented each of which is applicable to one specific network context. The selection of schemes to be practically deployed is dependent on the network characteristics and the system requirements. On environmental noise awareness, noise type- and SNR classification based ASR framework is presented. In this paper it has been experimentally justified that it is feasible to exploit context information to improve the robustness of ASR systems. Future work will include speaker context-awareness by applying the concept of multiple model framework and model adaptation to deal with speaker-related variations such as different age, gender and dialect.

8. Acknowledgements

This work has been supported by the FACE project, the CNTK project and the EU 6th framework project MAGNET.

9. References

[1] Furui, S., “Speech Recognition Technology in the Ubiquitous/Wearable Computing Environment”, ICASSP’00.

[2] Schilit, B., Adams, N. and Want, R., “Context-Aware Computing Applications”, *IEEE Workshop on Mobile Computing Systems and Applications*, December 1994.

[3] Dey, A.K. and Abowd, G.D., “Towards a Better Understanding of Context and Context-Awareness”, Technical Report, College of Computing, Georgia Institute of Technology, 1999.

[4] Abowd, G.D. et al., “Context-aware computing”, *IEEE Pervasive Computing*, 1 (3): 22 – 23, 2002.

[5] Chong, C.-Y. and Kumar, S.P., “Sensor Networks: Evolution, Opportunities, and Challenges”, *Proc. of The IEEE*, 91(8), 2003.

[6] Leiberman, H. and Selker, T., “Out of Context: Systems That Adapt To, and Learn From, Context”, *IBM Systems Journal* 39 (3-4): 617-632, 2000.

[7] Rose, R.C. et al., “On the Implementation of ASR Algorithms for Hand-Held Wireless Mobile Devices”, ICASSP’01.

[8] Raffaele Giaffreda et al., "D.6.1 Ambient Networks ContextWare", the EU 6th framework project Ambient Networks, Jan. 2005. http://www.ambient-networks.org/

[9] Yu Shi and Eric Chang, “Studies in Massively Speaker-Specific Speech Recognition”, ICASSP’04.

[10] Akbacak, M. and Hansen, J.H.L., “Environmental Sniffing: Noise Knowledge Estimation for Robust Speech Systems”, ICASSP’03.

[11] Tan, Z.-H., Dalsgaard, P. and Lindberg, B., “Half Frame-Rate Front-End and a Frame-Rate Switching Scheme for Distributed Speech Recognition”, submitted to IEEE MMSP’05.

[12] Goyal, V.K., “Multiple Description Coding: Compression Meets the Network”, *IEEE Signal Processing Magazine*, 18 (5), 2001.

[13] James, A.B., and Milner, B.P., “An Analysis of Interleavers for Robust Speech Recognition in Burst-Like Packet Loss”, ICASSP’04.

[14] Tan, Z.-H., Dalsgaard, P. and Lindberg, B., “A Subvector-Based Error Concealment Algorithm for Speech Recognition over Mobile Networks”, ICASSP’04.

[15] Pearce, D. and Hirsch, H., “The Aurora Experimental Framework for the Performance Evaluation of Speech Recognition Systems under Noisy Conditions”, ICSLP’00.

[16] Xu, H., Tan, Z.-H., Dalsgaard, P. and Lindberg, B., “Robust Speech Recognition Based on Noise and SNR Classification – a Multiple-Model Framework”, Interspeech’2005.

[17] Sagayama, S., Yamaguchi, Y. and Takahashi, S., “Jacobian Adaptation of Noisy Speech Models”, IEEE ASRU’97.

[18] Z.-H. Tan, P. Dalsgaard and B. Lindberg, “Automatic speech recognition over error-prone wireless networks,” *Speech Communication*, in press, 2005.