SPECTRAL SUBTRACTION IN ACOUSTIC MODELING FOR EMBEDDED SPEECH RECOGNITION

Jozef Ivanecký, Murat Deviren , Siegfried Kunzmann

IBM AIM, European Voice Technology Development
Schönaicher-Str. 220, 71032 Böblingen, Germany
{ivanecky, mdeviren, kunzmann}@de.ibm.com

ABSTRACT

Speech recognition on embedded systems is becoming as popular and important as the one on "real" computers. One of the main challenges in embedded speech recognition is the limitation on hardware resources. While in server based applications the CPU and memory are practically unlimited, in embedded world it is necessary to develop special techniques to achieve comparable recognition rates. Furthermore, in embedded applications we usually have to cope with more severe environment conditions. In this paper we describe our efforts on alternative training strategies for noisy speech recognition in embedded systems with the aim of achieving a higher degree of robustness under noise. We focus on the use of spectral subtraction technique in decoding and especially in the acoustic model training. We give an overview on the benefits of spectral subtraction for speech recognition in different noise conditions considering the recognition rate. Next, we explore different strategies to use spectral subtraction in acoustic modeling as a part of front-end feature vector computation. We evaluate different training scenarios based on overall recognition accuracy under different environment conditions.

1. INTRODUCTION

Today, on demand voice access to information is becoming more and more common in real applications. While server and desktop systems have reached a certain maturity in noise robustness, where only weak CPU and memory usage constraints exist, the situation is completely different for systems supposed to run on small devices like PDAs, or mobile phones. In this case, both restrictive CPU and memory requirements, and the need to run in real-time still result in high error rates. Besides, typical environment conditions for such systems include a noisy acoustic channel with background noise of many different characteristics. This is an additional challenge for the creation of small acoustic models that are capable to recognize the speech in different environments.

In this paper we describe our efforts for noisy speech recognition in embedded systems with the aim of achieving a higher degree of robustness under noise. We focus on the use of spectral subtraction technique in decoding but also and more importantly in acoustic model training. We explore different strategies to use spectral subtraction in acoustic modeling as a part of front-end feature vector computation. We evaluate different training and testing scenarios.

We first describe our speech recognition system and evaluation setup designed for an in-car embedded system in Section 2. The remainder of the paper is organized as follows: In Section 3 we give a brief overview over the recognition in noisy environment without and with spectral subtraction. In Section 4 we focus on the spectral subtraction in acoustic model training. Section 5 describes the experiments and a brief summary is given in Section 6.

2. IBM EMBEDDED SPEECH RECOGNITION SYSTEM

The design of an embedded speech recognizer has to deal with limited computational resources, both in terms of CPU power and memory capacity that today's embedded devices can offer. While some applications may run entirely on the local device and therefore require a relatively compact acoustic model, others may defer parts of the recognition process to a recognition server, which requires compatibility of at least the client's and server's acoustic front–end.

The latter is ensured by the use of a standard acoustic front–end, that computes 13 Mel Frequency Cepstrum Coefficients (MFCC) every 15 milliseconds. Utterance based cepstral mean subtraction and C0 normalization are applied to compensate for the acoustic channel and the first and second order delta coefficients

are computed to capture the temporal dynamics of the speech signal.

Recognizer training comprises the definition of a suitable HMM inventory and the determination of the HMM parameters. For that purpose, the training data is viterbi–aligned against its transcription in order to obtain an allophonic label for each feature vector.

Context dependent non cross–word triphone HMMs are obtained from the leaves of a decision network [1] that is constructed by asking binary questions about the phonetic context P_i for each feature vector, $i = -1, \ldots, 1$. These questions are of the form:"Is the phone in position i in the subset S_j?", and the subsets are derived from meaningful phone classifications commonly used in speech analysis. Finally, the data at each leaf of the network is used in a k–means procedure to obtain initial output probabilities whose parameters are then refined by running a few iterations of the forward–backward algorithm.

The k–means procedure follows a simple rule of thumb and equally distributes a fixed number Gaussian mixture components across the HMM states. Usually, in a highly dynamic and heterogeneous environment, an increased total number of Gaussians can significantly improve the recognition accuracy. However, this is not feasible for applications that have to deal with a limited amount of memory, and therefore the determination of an appropriate acoustic model size is of particular importance.

In this paper we train and evaluate our system for the German language considering an in car application. The training data consists of recordings in various noisy conditions with different types of recording devices, i.e. close talk microphone, far field microphone etc.. Our test set is designed with respect to the conditions in todays cars. The testing data is recorded from 42 different speakers in 4 different speeds: 0 km/h, around 40 km/h in the city, around 80 km/h outside the city and finally around 120 km/h on the highway. Each speaker recorded several hundred utterances covering different situations in the car. The data is tested against 5 different grammars like digit sequences (GR1), cell phone command and control (GR2), car command and control (GR3), navigation (GR4) and spelling (GR5). The standard test results that we will use later as the reference system are in Table 1. All the testing is done with the IBM's Embedded Speech Engine (ESE); see [2] for an overview on design issues and performance.

From the results in Table 1 it is clear that with the standard techniques the usage of the speech recognition in the car is limited especially with increasing background noise.

	Speed1	Speed2	Speed3	Speed4	Overall
GR1	1.25%	2.88%	3.85%	10.37%	3.74%
GR2	4.28%	5.15%	10.74%	18.49%	8.44%
GR3	1.15%	1.63%	2.46%	8.60%	2.64%
GR4	0.82%	1.92%	3.55%	20.28%	4.41%
GR5	12.10%	17.45%	26.60%	44.76%	22.35%

Table 1. Word error rate for the recognition with the standard build without the spectral subtraction.

3. SPECTRAL SUBTRACTION

Today's main challenge for the embedded speech recognition system is the automotive industry. Dialing the telephone number or controlling the navigation system are among the tasks that needs to be done by voice, especially if the car is already in move. While for speeds up to 60 km/h the noise in the car is relatively low, with higher speed we have to deal with SNR levels of 0–5 dB or even worse. Another challenge for in car systems is the recording device which is usually a far field microphone. The background noise interference is much more stronger when far field microphones are used instead of close talk microphones.

To deal with these problems and increase the recognition accuracy under severe noisy conditions we need to apply a noise reduction technique. Whereas in server based applications we can make use of complex algorithms for speech enhancement and/or noise modelling, in embedded domain we have to keep the CPU and memory consumption to a minimum. Therefore we need a computationally inexpensive but efficient algorithm for noise reduction. The well-known spectral subtraction(SS) method [3] belongs to this category.

Spectral subtraction is a speech enhancement algorithm which directly subtracts an estimate of a noise frequency spectrum from a noisy speech frequency spectrum. If we know what the noise looks like, then it will remove the noise, and leave just the speech. Unfortunately, we never know exactly what the spectrum of a noise source looks like, so we have to estimate it. The quality of the speech enhancement depends entirely on the estimate of the noise spectrum. The noise spectrum is estimated during the non speech period of the utterance. To determine when the noise spectrum should be adapted, the information from speech silence detector is used.

Our first approach is to apply SS on our test data before the recognition process and keep the acoustic model untouched. The results on reference system with spectral subtraction are in Table 2.

The positive influence of spectral subtraction on the

	Speed1	Speed2	Speed3	Speed4	Overall
GR1	1.22%	2.80%	3.74%	10.08%	3.64%
GR2	3.68%	4.43%	9.24%	15.90%	7.26%
GR3	1.05%	1.48%	2.24%	7.82%	2.40%
GR4	0.57%	1.34%	2.47%	14.13%	3.07%
GR5	11.43%	16.49%	25.14%	42.30%	21.12%

Table 2. Word error rate for the recognition with the standard build and the spectral subtraction switched on.

speech recognition accuracy is visible. But results are not always as good as we expect. The first reason is well-known problem of the estimation of noise spectrum. The second one is related to mismatch between the acoustic model training data and the test data. The acoustic model is trained on clean and noisy data but not on the influence of SS on the audio. It is a well-known phenomenon that after SS a musical noise resides on the audio signal. This comes from the nature of the algorithm and it is a reasonable trade-off for the low resources required.

4. SPECTRAL SUBTRACTION IN THE ACOUSTIC MODELING

The easiest way to adapt the acoustic model to the spectrally subtracted input speech signal would be the acoustic training with both kind of cepstra — normal cepstra and spectrally subtracted cepstra — and increased total number of Gaussians. From the limitation for acoustic model described above it is clear that this is not acceptable solution. To find out the best approach within the given limitation we needed to run several experiments:

- Acoustic training contains doubled number of training data. 50% of them are spectrally subtracted. The maximum number of Gaussians is not changed. During the decoding we need to test the performance for:
  - Standard mode without spectral subtraction
  - Mode with spectral subtraction on
- Acoustic training contains standard number of data. All the data are spectrally subtracted. The maximum number of Gaussian is the same as in all other training scenarios. During the decoding we need to test the performance for:
  - Standard mode without spectral subtraction
  - Mode with spectral subtraction on

	Speed1	Speed2	Speed3	Speed4	Overall
GR1	1.25%	2.99%	4.58%	12.22%	4.25%
GR2	4.38%	5.22%	6.76%	23.56%	7.91%
GR3	1.07%	1.48%	2.85%	11.26%	3.04%
GR4	0.82%	1.57%	2.30%	11.52%	2.84%
GR5	12.60%	19.14%	29.03%	49.86%	24.38%

Table 3. Word error rate for the recognition with the build with dual cepstra and the spectral subtraction switched off.

	Speed1	Speed2	Speed3	Speed4	Overall
GR1	1.17%	3.20%	4.08%	9.77%	3.81%
GR2	4.31%	5.18%	5.34%	13.81%	6.14%
GR3	1.05%	1.60%	2.35%	7.99%	2.49%
GR4	0.85%	1.52%	1.66%	7.71%	2.16%
GR5	12.56%	17.66%	25.87%	44.00%	22.20%

Table 4. Word error rate for the recognition with the build with dual cepstra and the spectral subtraction switched on.

To create the spectrally subtracted training cepstra we used the same PCM data as for standard cepstra. The allophonic labels were reused also from the standard cepstra. The training data were used for training of two new acoustic models.

5. EXPERIMENTS

In the first acoustic model we used both kind of cepstra to preserve the performance in the mode without spectral subtraction. The amount of the training data was doubled but all other build settings were the same as in standard build. In Table 3 we can see the result for the decoding in the standard mode.

Comparing to the reference build there is a small degradation which was expected due to reduced number of Gaussian covering standard cepstra. Situation in spectral subtracted mode is different. The results are in Table 5.

The results for high SNR test data are worse then for reference build with spectrally subtracted cepstra, but for the lower SNR we can see the significant improvement against the reference build with spectral subtraction. For 80 km/h the relative improvement is 24% and for 120 km/h the improvement in word error rate is 18%.

The interesting results were obtained for the second acoustic model where only spectral subtracted cepstra were used in the acoustic training. The first experiment was done again in the standard mode without spectral

	Speed1	Speed2	Speed3	Speed4	Overall
GR1	1.01%	2.73%	3.91%	11.13%	3.75%
GR2	4.38%	5.18%	10.97%	21.22%	8.90%
GR3	1.05%	1.58%	2.55%	8.99%	2.67%
GR4	0.78%	1.43%	1.88%	12.11%	2.73%
GR5	12.06%	17.78%	27.55%	46.78%	22.98%

Table 5. Word error rate for the recognition with the build with spectral subtracted cepstra only and the spectral subtraction switched off.

	Speed1	Speed2	Speed3	Speed4	Overall
GR1	1.10%	3.07%	3.77%	10.05%	3.69%
GR2	4.23%	4.87%	4.74%	15.75%	6.10%
GR3	1.05%	1.60%	2.41%	6.71%	2.35%
GR4	0.83%	1.43%	1.45%	10.01%	2.35%
GR5	12.47%	17.37%	25.04%	39.19%	21.23%

Table 6. Word error rate for the recognition with the build with spectral subtracted cepstra only and the spectral subtraction switched on.

subtraction. The results are in Table 5.

We expected a similar or bigger degradation for the standard mode as in the first build. But as is showed in the table in this case the results are comparable with the reference build and in average are even slightly better. Such results were unexpected and we do not have explanation for it.

In the last experiment we tested the build with spectral subtracted cepstra only with the spectral subtraction during the decoding. The results are comparable to the build with both kind of cepstra. For high SNR the results are worse then in case of reference build, but slightly better then for dual system. In case of low SNR we achieved similar improvement as in the build with the dual cepstra. In this case the results are worse then expected.

On the Figure 1 are averaged result for the reference build without spectral subtraction and all the 3 build with the spectral subtraction during the decoding. From the results is clear that spectral subtraction can have significant influence for the speech recognition in the noisy environment. The best improvement is achieved if spectral subtraction is used not only in the speech decoding but also in the acoustic model training.

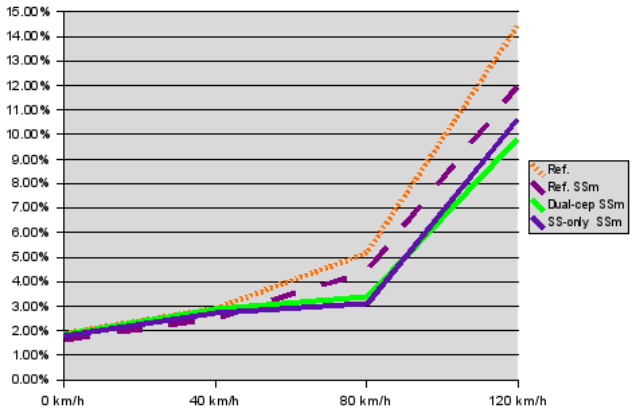

Fig. 1. The averaged word error rate for the reference system (Ref.), reference system with the SS decoding (Ref. SSm), dual cepstra system with the SS decoding (Dual-cep SSm) and SS cepstra only system with the SS decoding (SS-only SSm).

6. SUMMARY

In this paper we described various aspects of the spectral subtraction in the speech recognition process. We explored the usage of the spectral subtraction in the decoding step and designed the next steps toward the usage of the spectral subtraction also in the training process. Experiments with the spectral subtraction in the acoustic modeling demonstrated the feasibility of our approach, but at the same time unveiled the need for further research in very noisy environment speech detection in order to better apply the spectral subtraction in the embedded speech recognition systems without an unwanted decrease in accuracy.

7. REFERENCES

[1] L. Bahl, P. de Souza, P. Gopalakrishnan, D. Nahamoo, M. Picheny. Context-dependent Vector Quantization for Continuous Speech Recognition. In Proc. of the IEEE Int. Conference on Acoustics, Speech, and Signal Processing, Minneapolis, 1993.

[2] T. Beran, V. Bergl, R. Hampl, P. Krbec, Jan Šedivý, B. Tydlitát, J. Vopička. Embedded ViaVoice. In Proc. of TSD 2004, Brno, 2004.

[3] Boll SF. Suppression of Acoustic Noise in Speech using Spectral Subtraction. In Proc. of the IEEE Trans. Acoust., Speech, and Signal Proc., Vol. A, pp. 113-120, 1979.