THE ISL EVALUATION SYSTEM FOR VERBMOBIL-II

Hagen Soltau, Thomas Schaaf, Florian Metze, and Alex Waibel

Interactive Systems Laboratories
University of Karlsruhe (Germany), Carnegie Mellon University (USA)
{soltau,tschaaf,metze,waibel}@ira.uka.de

ABSTRACT

This paper describes the 2000 ISL large vocabulary speech recognition system for fast decoding of conversational speech which was used in the German Verbmobil-II project. The challenge of this task is to build robust acoustic models to handle different dialects, spontaneous effects, and crosstalk as occur in conversational speech. We present speaker incremental normalization and adaptation experiments close to real-time constraints. To reduce the number of consequential errors caused by out-of-vocabulary words (OOV), we conducted filler-model experiments to handle unknown proper names. The overall improvements from 1998 to 2000 resulted in a word error reduction from 40% to 17% on our development test set.

1. INTRODUCTION

Verbmobil is a long-term research project aimed at automatic speech-to-speech translation between German, English, and Japanese. Several universities and industry partners are involved in this project. In the first phase of Verbmobil (VM-I), the domain was very limited and the speaking style was cooperative with less spontaneous effects. In the second phase of Verbmobil (VM-II), the data became more realistic with a couple of spontaneous effects and crosstalk. As it is shown in table 1, the scenario of VM-II was extended by a travel domain resulting in much higher perplexity.

	VM-I	VM-II
Speaking style	cooperative	conversational
Crosstalk	no	yes
Vocabulary	2500	10000
Perplexity	38	112[1]
OOV	1.8%	2.4%
Speech Data	34h	62h
LM Data	300k	670k

Table 1: task description

In this paper, we describe our work on developing a fast, and robust speech recognizer for the Verbmobil-II task. After a brief overview of our system, we will give details about our experiments on building robust acoustic models for conversational speech, namely improved noise modeling, robust channel normalization, speaker incremental feature space adaptation during training and decoding, and OOV detection. Additionally, we will report some results of speeding up our system using the Bucket Box Intersection Algorithm (BBI) [4] and Phoneme Lookaheads. We used two development test sets *dev98* and *dev99* for the experiments. The final Verbmobil evaluation was carried out on the *eval00* test set (see table 2). The systems of each participant were evaluated by a neutral site at University of Braunschweig [8].

test set	dialogues	turns	duration
dev98	20	763	60min
dev99	15	538	53min
eval00	15	774	61min

Table 2: dev/eval test sets

2. ACOUSTIC MODELING

Starting from a context independent system, we built a phonetically tied system and train models for all seen quintphones. We use a likelihood criterion on a cross-validation set (round robin) to split the models in a top-down clustering procedure. Besides context questions, we use word boundary questions to cover coarticulatory effects. In the final system, we used 3500 speech states, each modeled by a mixture of 48 gaussians. To model the acoustic space with such a huge number of gaussians (168k gaussians in total) is even useful[2] for building a fast system, since score computations can be better pruned during decoding using gaussian space partition methods or phonetic lookaheads instead of just training small acoustic models. To illustrate this, we built three systems with different model sizes and used a BBI tree [4] to speed up the systems to achieve similar real time factors (table 3).

Usually, we use fixed frame/phone alignments (generated by a previous system) to accumulate sufficient statistics instead of performing full viterbi or Baum-Welch training in each iteration. We didn't seen any performance

[1] measured on eval00 test set

[2] given enough training data available

0-7803-7041-4/01/$10.00 ©2001 IEEE

nr. of speech states	error rate	real time
1666	23.4%	4.3
2777	22.6%	4.6
4300	22.9%	5.1

Table 3: model size vs. search effort (results on dev98)

degradation caused by this approximation, so that we used this faster training procedure for all experiments.

2.1. Preprocessing

The feature extraction in our system is based on mel-filtered cepstral coefficients with their delta, and delta-delta's. For channel normalization, we apply speaker incremental, cepstral mean subtraction (CMS) with exponential history weighting. In table 4, we summarize our experiments with cepstral variance normalization (CVN) and linear discriminant analysis (LDA). For the non-LDA systems, the variance normalization gave us a gain of 0.7% absolute. After we applied the LDA transformation, CVN is no longer effective. Indeed, the LDA performs a kind of static variance compensation, e.g. the $C0$ values (which have typically high variances) will be scaled down by a factor of 10. Another interesting observation is, that the gain of the LDA transformation is above 2.1% absolute, although we cut the LDA feature vectors from 39 coefficients to 32.

CVN	LDA gain
no	23.4% → 21.3%
yes	22.7% → 21.4%

Table 4: Preprocessing (results on dev99)

For all these experiments, we apply Vocal Tract Length Normalization (VTLN) in a speaker incremental, exponential history weighted way as we mentioned it for CMS. To avoid several search passes during decoding, we favor a delayed reestimation of the warp factor after decoding of the current utterance of the speaker. We got almost the same gain compared to a complete speaker based estimation.

2.2. Semi-tied covariances (STC)

Since, a linear transformation that maximize the ratio of between group to within group variances, as a LDA does, is only optimal for problems with same group variances, we investigated the use of linear feature transformations which directly maximize the likelihood [6, 7].

After the encouraging results with LDA, we built systems using LDA and STC to see if there are additional gains available. Even with a LDA frontend, we achieved a 8.7% error reduction with STC (see table 5). Additionally, we examined how to structure and tie the covariances. These experiments were performed during an early development stage, where we used the *dev*98 test set (table 6).

Preprocessing	STC gain
no LDA	22.3% → 19.9%
LDA	20.8% → 19.0%

Table 5: Linear discriminant analysis and semi-tied covariances (results on dev99)

nr. of matrices	type	error rate
Baseline		22.6%
global	block	22.3%
three classes	block	22.2%
phone dep.	block	21.5%
state dep.	block	21.8%
global	full	21.5%
phone dep.	full	20.8%

Table 6: structuring semi tied covariances (results on dev98)

2.3. Human and Nonhuman Noise Modeling

We model in our system mainly two kinds of noises. For the language model, we use two vocabulary entries to cover human and nonhuman effects. Each of these models has several pronunciation variants to cover different acoustic surfaces. Additionally, we build context decision trees for the human noises. The question is now, how to train these models. Fortunately, the word level transcriptions in Verbmobil contain some noises. We used these transcriptions to train seed models. After that, we retranscribed automatically the data by computing forced alignments through flexible HMM's as shown in figure 1. This was originally proposed by Michael Finke in [3].

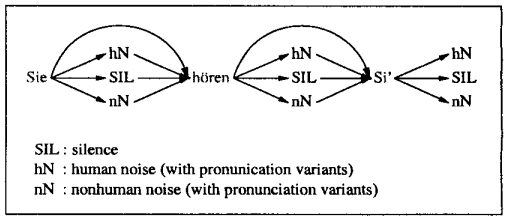

Figure 1: dynamic noise transcription

The results of different transcription strategies can be seen in table 7. We got an improvement of 1.4% absolute by this noise model training. Another problem of the Verbmobil data is that some speech is corrupted by crosstalk from a different channel. To avoid that the speech models are trained on wrong crosstalk samples, we trained a special crosstalk model, that we used as a filler model for the flexible transcription alignment approach.

optional words	error rate
silence	24.0%
+ human, nonhuman noises	22.6%
+ mumble, <eh>, , <hm>	22.7%

Table 7: dynamic noise transcription, error rates on dev98

2.4. Feature Space Adaptation

There are mainly two problems if we apply speaker incremental MLLR adaptation for a real time system. After each adaptation step, we have to transform all gaussians. For a large system with 160k gaussians, this need approx. 5 seconds on $SUN2/300$ machine. Now, the average duration of one utterance is nearby 5 seconds, this means, we need almost one real time factor to apply the transformation, even in a transformation on demand approach during decoding. Additionally, the BBI algorithm that we use for gaussian selection to speedup the score computation, computes partitions of the model space. A transformation of the model space causes a mismatch of the underlying BBI tree, resulting in wrong gaussians selection. Instead of transforming the models, we therefore compute a linear transform for the feature space to avoid the problems mentioned above. The optimization criterion base on a normalized likelihood function to incorporate the Jacobi determinant [5]. We apply this technique also during training to normalize speaker and channel effects across the dialogs. This can be seen as a very efficient speaker adaptive training variant [1].

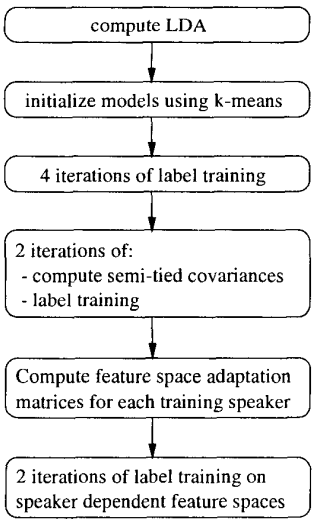

Figure 2: training scheme using fixed labels with semi-tied covariances and feature space adaptation

The general training scheme of the context dependent models is outlined in figure 2. After four iterations of label training to generate seed models, we train semi-tied covariances for two iterations. Then we go into the optimal feature space and estimate adaptation matrices for each training speaker. Two training iterations using the new speaker dependent feature spaces follows.

During decoding, we use some training data from female and male speech to enhance the robustness of the adapation parameter estimation [2]. We combine the statistics from the test speaker with either the female or male training accumulators and then estimate the adaptation matrices. Through this techniques, we achieved a word error reduction of 8.6% (table 8). The baseline system already use CMS, LDA, semi-tied covariances, and vocal tract length normalization.

System	error rate
no adaptation	25.7%
feature space adaptation	23.5%

Table 8: adaptation results, error rates on eval00

2.5. OOV Detection

To reduce the number of indirect errors due to OOV words, we created special filler models, both in the acoustic and language models. For this purpose, we trained a global phoneme using the data from all vowels and consonants to cover unknown phone sequences. Additionally, we used some frequent vowel/consonants combinations to integrate phonotactic knowledge. As shown in table 9, we achieved a word error reduction of 0.7%.

System	w/o UNK mapping	with mapping[3]
Baseline	23.8%	23.8%
filler models	23.6%	23.1%

Table 9: oov detection, results on dev98

In figure 3 we show the overall improvements during the last three years. In 1998, we started with error rates around 40%. By now, we have a system with a error rate of 16% on the $dev99$ test set.

3. DECODING

The decoder works in three passes. In the first pass, a tree structured vocabulary is used to generate lists of starting words for each word using LM unigram lookaheads and delayed approximative trigrams. In the second pass, a flat organized vocabulary is used to incorporate exact bigrams. The resulting lattice will be rescored using long-span language models in the last pass. To avoid additional acoustic score computations, a score cache is used for the different

[3]In the official evaluation, unknown words of the reference strings were mapped to a special lexical entry $< UNK >$.

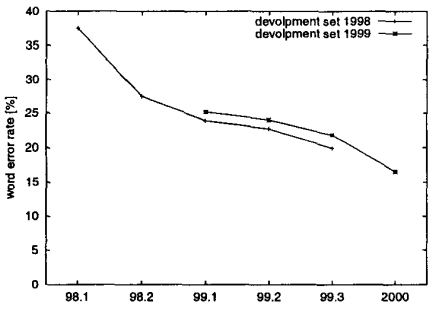

Figure 3: error reductions from 1998 to 2000

passes. The acoustic score computation is speeded up by the BBI algorithm, which select gaussians close to the incoming feature vector. A phonetic lookahead strategy is used to reduce the search space. The lookahead models base on a small context independent system. Additionally, we use a maximum approximation for the mixtures of gaussians. The effect of speeding up vs. error rate is shown in table 10. The final evaluation system runs in 1.3 realtime on a PentiumIII-600 machine.

System	error rate	real time
baseline	16.2%	29.3
tighter beams	16.5%	12.9
max approx.	16.5%	5.1
bbi	16.7%	1.8
Lookaheads	16.8%	1.3

Table 10: Speeding up the system (dev99 on PentiumIII-600)

The computational efforts for each search pass are listed in table 11. We need just 0.19 real time factors (rtf) for speaker normalization and feature space adaptation, because this is performed in a delayed incremental way.

Search pass	real time
preprocessing	0.04
tree	0.85
flat	0.15
rescoring	0.02
vtln	0.09
adaptation	0.10
total	1.25

Table 11: cpu-usage for each search pass (PentiumIII-600)

On the final evaluation test set, we achieved a error rate of 25.2% at 1.5 rtf on a PentiumIII-600 (2.1 on SUN). The speed up of the system from 20 rtf to 1.5 rtf caused approx. 10% error increase on the eval00 test set.

4. SUMMARY

In this paper, we have described our efforts to build a fast and accurate system for decoding conversational speech. Substantial error reductions were achieved using speaker normalization and adaptation algorithms under real-time constrains. We examined methods to train robust acoustic and language models to cover human and nonhuman noise effects. Acoustic and language models were improved to handle unknown proper names. The system achieved best word error rates in the Verbmobil evaluations.

5. ACKNOWLEDGMENTS

The authors wish to thank Jürgen Fritsch and Michael Finke for many fruitful discussions. This work is partly funded by grant 413-4001-01IV101S3 from the German Ministry of Science and Technology (BMBF) as a part of the VERBMOBIL project.

6. REFERENCES

[1] T. Anastasakos, J. McDonough, R. Schwartz, and J. Makhoul. A compact model for speaker-adaptive training. In *Proceedings of the ICSLP*, Philadelphia, USA, 1996.

[2] W. Byrne and A. Gunawardana. Discounted likelihood lineare regression for rapid adaptation. In *Proceedings of the Eurospeech*, Budapest, Hungary, 1999.

[3] M. Finke and A. Waibel. Flexible transcription aligment. In *Proceedings of the Automatic Speech Recognition and Understanding (ASRU).*

[4] J. Fritsch and I. Rogina. The bucket box intersection (BBI) algorithm for fast approximative evaluation of diagonal mixture gaussians. In *Proceedings of the ICASSP*, Atlanta, USA, 1996.

[5] M.J.F. Gales. Maximum likelihood linear transformations for HMM-based speech recognition. Technical report, Cambridge University, England, 1997.

[6] M.J.F. Gales. Semi-Tied Full-Covariance matrices for hidden markov models. Technical report, Cambridge University, England, 1997.

[7] N. Kumar. *Investigation of Silicion-Auditiory Models and Generalizations of Linear Discriminant Analysis.* PhD thesis, Johns Hopkins University, Baltimore, USA, 1997.

[8] M.Malenke, M. Bäumler, and E. Paulus. Speech recognition performance asssessment. In *Verbmobil: Foundations of Speech-to-Speech Translation*, Springer-Verlag, 2000.

[9] I. Rogina. Automatic architecture design by likelohood-based context clustering with crossvalidation. In *Proceedings of the Eurospeech*, Rhodos, Greece, 1997.

[10] M. Woszczyna. *Fast Speaker Independent Large Vocabulary Continuous Speech Recognition.* PhD thesis, University of Karlsruhe, Germany, 1998.