THE 2003 ISL RICH TRANSCRIPTION SYSTEM FOR CONVERSATIONAL TELEPHONY SPEECH

Hagen Soltau, Hua Yu, Florian Metze, Christian Fügen, Qin Jin, Szu-Chen Jou

Interactive Systems Laboratories
University of Karlsruhe (Germany), Carnegie Mellon University (USA)
{soltau,metze,fuegen}@ira.uka.de, {hyu,qjin,scjou}@cs.cmu.edu

ABSTRACT

This paper describes the ISL large vocabulary conversational telephony speech recognition system, which was tested in NIST's RT-03S ("Switchboard") evaluation. We present our experiments on improving preprocessing, acoustic modelling, and language modelling. The system features phone dependent semi-tied full covariances, semi-tied clustering of septa-phones, clustering across phones, feature adaptive training, robust estimation of VTLN and MLLR, as well as context dependent interpolation of language models. We present detailed results for each stage of our multi-pass transcription scheme. System development started in 2002 with an error rate of 35.1% on our internal 1h development set. The final system performed at WER 21.8%, a 38% relative improvement. The error rate on the RT-03 CTS evaluation set is 23.4%.

1. INTRODUCTION

Recognition of conversational telephony speech is a challenging task, both for acoustic and language modelling. Under-articulated speech causes a mismatch with the pronunciation dictionary and the acoustic models. Consequently, normalisation and adaptation techniques are heavily used during training and decoding. Additionally, sloppy speech makes it hard to train appropriate language models. Furthermore, automatic segmentation was required for the first time in the HUB-5 evaluation series.

We started by reviving ISL's 1997 SWB system [1]. This system runs in several passes estimating normalisation and adaptation parameters and obtained a top rank in the 1997 evaluation. Running this system as-is on the 2001 evaluation set, we achieved an error rate of 34.8%. Our 1997 system is significantly behind the top systems in the 2001 SWB evaluation [4] demonstrating that systematic improvements were made in the last years in the ASR community.

We used two test sets (table 1) for development purposes. Dev01 is a 1h subset designed to have a similar error rate as the full Eval01 set (35.1% and 34.8%). Tests with automatic segmentation were performed on the "Dry-run" data. Unless otherwise stated, the reported error rates are based on Dev01. Since several setups for different experiments were used during system development, the stated error rates do not decrease monotonously and the results need to be viewed with respect to the corresponding baseline.

The paper is organised as follows: First, we present experiments for acoustic modelling including front-end and segmentation. Next, we describe our language models and the decoding strategy and present results for each system stage. In the final section, we describe experiments to reduce the decoding time.

	subset from	segmentation
Dev01	1h from Eval01	manual
Dry-run	1h from Eval02	automatic

Table 1: Development sets.

2. ACOUSTIC MODELLING

Acoustic models were trained by merging three corpora: 265h of SWB+Callhome, 32h of Cellphone data, and 65h SWB-2 data transcribed by CTRAN. The Cellphone data were weighted by a factor of 3, and a factor of 2 was used for the CTRAN data. The original ISIP training transcripts were used for the SWB data. The training dictionary was derived from CMUdict and on average contains nearly 2 pronunciation variants per base-form.

Since we started with a new training environment from scratch, we performed several steps to clean-up the database. By discarding all training segments containing one word only, an error reduction from 37.1% to 36.4% was obtained. Furthermore, we limited the segment boundaries to max. 15 frames of silence only. Some SWB conversations contain segments without any audio signal. These zero-energy frames lead to extreme likelihoods, in particular in combination with feature space adaptation. Discarding these frames by using a zero-crossing feature resulted in an improvement from 33.4% to 32.8%. Additionally, segments with poor likelihoods were removed as well.

2.1. Preprocessing

The 5min excerpts from the conversations were segmented into smaller chunks before decoding. The segmentation works in two phases. An initial energy-based raw segmentation with three categories (speech, non-speech, unsure) is used to bootstrap GMMs for speech and non-speech. These GMMs are used to re-segment the unsure parts. Finally, a smoothing process is conducted to join adjacent chunks.

As shown in table 2, segmentation error rate and word error rate are not necessarily correlated. If segmentation

system	seg. error	WER
manual	-	41.1%
automatic	14.3%	41.9%
automatic	9.7%	43.0%

Table 2: Segmentation on Dry-run.

is optimised for ASR, only a minor degradation (0.8%) is observed when comparing to manual segmentation.

The front-end is based on 13 mel-filtered cepstral coefficients per frame, applying conversation wide cepstral mean subtraction. Incorporating context information by concatenating 11 frames gave significantly better results than a Δ-based approach. Furthermore, per-speaker variance normalisation gave an additional 0.9% gain.

front-end	WER
Δ's+$\Delta\Delta$'s+LDA	39.7%
frame stacking+LDA	38.5%
+ CVN	37.6%

Table 3: Front-end improvements.

Our VTLN procedure is based on a grid search maximising the likelihood for voiced sounds. Traditionally, warping factors are estimated with fixed CMS/CVN which introduces inconsistencies. We overcame this drawback by an interleaved estimation of all parameters, e.g. the likelihood for a given warping factor is computed with the correct cepstral mean and variance normalisation. This makes it desirable to use a more efficient search method. The new VTLN procedure is, therefore, based on Brent search. The interleaved estimation yielded an improvement from 33.2% to 32.4%. The final feature vectors are transformed by an LDA using the context dependent states as classes and the dimension is reduced from 143 (13 MFCC's * 11 frames) to 42.

2.2. Training procedure

The training procedure is based on fixed state alignments. In our experiment, the alignments were generated with a *small* context dependent system. As shown in table 4, these labels are significantly better than the labels generated with the full setup[1]. We attribute this result to the better generalisation capability. Moreover, the fixed alignment approach using a small setup outperforms both viterbi and forward/backward training. Generating once a set of frame/state alignments and keeping them fixed over the training iterations reduces the training time drastically compared to traditional training. This will become an important issue once 2000h of Fisher training data are available.

Our traditional training procedure bootstrapped the models with the K-means algorithm. Alternatively, we implemented an "incremental growing of Gaussians"- procedure. Starting with one component per state, the Gaussians will be splitted according to the largest covariances. A occupancy threshold is used to deactivate "dead" Gaussians. The training consists of 7 big iterations with parameter doubling. After each big iteration, three "small" re-estimation steps are preformed without splitting. This strategy is particularly advantageous for the 10000x32 setup (see table 5, where the models consists of 10000 states with 32 Gaussians. The final models have 288000 Gaussians due to the integrated pruning. Combining the "incremental growing" strategy with the fixed alignments leads to a very time and memory efficient training: The preprocessed data can be organised per context dependent HMM state. Therefore, the training can be parallelised according to the states instead of the utterances as usual. This reduces file-IO drastically since the data needs to be loaded only once for all iterations.

[1]The small setup uses 7% of the parameters that the full setup occupies.

alignment	WER
full fwd/bwd	33.1%
viterbi	33.3%
labels with full setup	33.5%
labels with small setup	32.7%

Table 4: Forced alignments.

method	10000x24	10000x32
init with k-means	33.8%	33.7%
incremental growing	33.1%	32.4%

Table 5: Training procedure.

2.3. Clustering

Context dependent models are created by an Entropy based clustering procedure. First, mixture weights for all polyphone models are trained, where the models rely on context independent codebooks. Questions about the phonetic context and the phone position are used to split the tree nodes. Extending the context from ± 2 to ± 3 yields a gain from 34.7% to 34.2%. The clustering is applied in two stages. In the first stage, context dependent states for the full model parameters are generated. In a second, extended, tree, a larger number of states is used for the mixture weights only. By doing so, we have one tree with 10000 codebooks and a second tree of 50000 distributions which rely on the first one. The second tree has only 5% more model parameters, but reduces the word error rate to 31.8% (from 32.8%, see table 6).

Traditional clustering grows one tree per context independent HMM state. As an alternative, we investigated across phone trees [9], offering better parameter sharing capabilities. The clustering procedure grows 6 trees only ("begin", "middle", and "end" for vowels and consonants). Clustering across phones implicitly models articulatory changes in sloppy speech.

acoustic models	# params	WER
10k cb's + 10k ds's	27.2M	32.8%
10k cb's + 50k ds's	28.5M	31.8%

Table 6: Two-level Clustering.

Clustering	dict.	train (66h)	train (180h)
traditional	multiple	34.4%	33.4%
across phones	multiple	33.9%	-
traditional	single	34.1%	-
across phones	single	33.1%	31.6%

Table 7: Clustering across Phones. Note the behaviour on dictionaries containing single or multiple pronunciation variants.

2.4. Semi-tied full covariances

Semi-tied full covariances (STC) [2] attempts to reduce the detrimental effects of diagonal covariance modelling. The STC parameters are trained on top of the LDA transform. The estimation procedure estimates all parameters, e.g. diagonal covariances and STC transforms, simultaneously resulting in a significant memory footprint. However, the containers for the statistics can be allocated on demand. In combination with the parallelisation over the HMM states, the memory footprint can be divided by the number of parallelised jobs. As shown in table 8, phone dependent STC classes do not work in combination with MLLR. Therefore, phone dependent STC classes are used only for the first, unadapted, decoding pass.

	w/o MLLR	with MLLR
no STC	36.7%	34.1%
global STC	33.7%	32.2%
phone STC	33.4%	33.1%

Table 8: Interaction of STC with MLLR.

STC training is also used for the test speakers. The global STC classes are re-estimated for each test speaker (additional to MLLR and FSA) and results in a minor improvement (26.8% $\rightarrow$ 26.6%).

2.5. Feature Space Adaptation

Feature space adaptation is used both in training and testing. The adaptive training is carried out per conversation side on top of the LDA/STC transforms. The VTLN factors are kept fixed during FSA re-estimation. A determinant constraint $|A| = 1$ is induced during the matrix estimation. As shown in table 9, FSA gives 1.1% improvement on top of all other normalisation and adaptation techniques. In contrast to FSA, where only one global matrix is used, MLLR makes use of a regression tree and the number of transforms depends on the adaptation data available.

setup	WER
VTLN,MLLR,STC	28.9%
+ FSA-SAT	27.8%

Table 9: Feature Space Adaptation.

2.6. MMIE training

The accumulation strategy for discriminative training is based on confusion networks [3]. First, lattices are generated using an uni-gram language model (LM). A downscaling of the LM scores is applied to "flatten" the word posterioris. Next, lattices are converted to confusion networks. The Forward/Backward procedure is applied on these networks. The word boundaries can therefore be adjusted during training in contrary to the "phone-marked lattice"-approach [8]. However, both accumulation procedures lead to the same results. A weighted ML and MMIE criterion is used to update the parameters. Only one iteration is used; the second iteration leads already to over-training on the full setup. The discriminative training leads to an error reduction from 28.3% to 27.6% on the full setup (LDA, VTLN, STC, FSA-SAT, MLLR, two-level clustering).

setup	ML	MMIE
small	41.9%	40.9%
full	28.3%	27.6%

Table 10: Discriminative Training.

3. LANGUAGE MODELLING

The search vocabulary contains 41000 base-forms and 96000 pronunciations selected from SWB, BN, and CNN corpora. The pronunciations were either taken from CMUdict or generated by Festival. Pronunciation probabilities were treated as penalties during decoding, and as real probabilities for confusion networks. The frequencies were generated from training labels. Three separate LMs were interpolated, whereby the weights depends on the predecessor words. As shown in table 11, the CNN LM did not improve the performance. Thus, the combined LM consists of 3gram SWB+ 5gram class SWB + 4gram BN. Apart from the first, unadapted, pass, all passes used the full interpolated LM during decoding and lattice generation.

3gram SWB	31.4%
+ 5gram class SWB	31.0%
+ 4gram BN	30.3%
+ 4gram CNN	30.5%

Table 11: Language Modelling, context dependent interpolation.

Pass 0	35.0%	Tree-150, MMIE, STC-50, smallLM
Pass 1/2	28.5%	Tree-150, ML , STC-1, VTLN, MLLR, bigLM
Pass 3/4	27.2%	Tree-150, MMIE, STC-1, VTLN, MLLR, FSA-SAT, bigLM
Pass 5	26.6%	Tree-6, ML, STC-1, FSA-SAT, bigLM, SPDict
Pass 6	26.2%	Tree-150, MMIE, STC-1, VTLN, MLLR, FSA-SAT, bigLM
Pass 7	26.4%	Tree-6, cross-adaptation
Pass 8	25.4%	Tree-150, cross-adaptation
Pass 9	24.7%	System Combination

Table 12: Decoding Passes (Tree-150= cluster per phone state, Tree-6=cluster across phones, STC-50= phone dependent classes, STC-1= global STC, results on Dry-run (automatic segmentation).

4. DECODING

The search engine is a one-pass decoder based on linguistic polymorphism [5]. The full LM history is conserved by linguistic instances of search nodes. Subgraph dominance is, therefore, exploited implicitly. The search network is based on a general graph structure, sharing roots and tails. Isomorphic subgraphs are merged via an iterative procedure reducing redundancies. Lattice nodes are created from the active search space by removing the LM information. Links are created during and after decoding. This allows to transfer as much information as possible from the active search space into the lattice.

The decoding passes are summarised in table 12. Pass 0 used phone dependent STC classes and the 3gram SWB LM only. Lattice based MLLR [7] is used to generate adapted models for the next pass. Passes 7 and 8 are used for cross-adaptation between the Tree-150 (traditional clustering) and Tree-6 (clustering across phones) setup. The effect of cross-adaption can be seen by comparing the passes 6 and 8 which use the same models. System combination uses a mixture of Rover and confusion network combination. We hereby fuse lattices from different stages into one single confusion network. Processing of the test data took about 190 times real-time on a 2.4GHz Pentium4 single CPU.

Since fast transcription systems receive increasing interest, we investigated the trade-off between speed and accuracy on the final, adapted, decoding pass. Using eval-mode search parameters gives an error rate of 24.2% on the Eval03 test set and a real time factor of 12. As shown in table 13, the decoding process can operate in real time with a moderate increase of search errors if appropriate beam settings are used.

Pruning parameter	RTF	WER
beam= 2.4 (eval mode)	12.0	24.2
beam= 1.5	4.7	24.6
beam= 1.1	1.4	26.0
+ transN=35	1.0	26.1
+ delayed interpol.	0.9	26.1

Table 13: Single adapted pass, RTF on P4 2.4Ghz, WER on Eval03.

5. SUMMARY

We described the development of ISL's 2003 transcription system for conversational telephony speech. The system achieved an error rate of 23.4% on the official RT-03 (Eval03) CTS test set. Starting last year with a WER of 35.1% on our development set, systematic improvements of acoustic and language modelling led to a WER of 21.8%.

Part of this work has been funded by the European Union as IST project No. IST-2000-28323 (FAME).

6. REFERENCES

[1] M. Finke, J. Fritsch, P. Geutner, K. Ries, and T. Zeppenfeld. The JanusRTk Switchboarch/Callhome 1997 Evaluation System. In *LVCSR Hub-5E Workshop*, Linthicum Heights, MD, USA, 1997.

[2] M.J.F. Gales. Semi-Tied Covariance Matrices for Hidden Markov Models. *IEEE Transactions on Speech and Audio Processing*, Vol. 2, May 1999.

[3] L. Mangu, E. Brill, and A. Stolcke. Finding Consensus among words: Lattice-based word error minimization. In *Proceedings of the Eurospeech*, Hungary, 1999.

[4] A. Martin and M. Przybocki. The 2001 NIST Evaluation for Recognition of Conversational Speech Over the Telephone. In *LVCSR Workshop*, Linthicum Heights, MD, USA, 2001.

[5] H. Soltau, F. Metze, C. Fügen, and A. Waibel. A One Pass Decoder based on polymorphic linguistic context assignment. In *Proceedings of the Automatic Speech and Recognition Workshop (ASRU)*, Trento, Italy, 2001.

[6] H. Soltau, T. Schaaf, F. Metze, and A. Waibel. The ISL Evaluation System for Verbmobil-II. In *Proceedings of the ICASSP*, Salt Lake City, USA, 2001.

[7] L. Uebel and P. Woodland. Improvements in linear transform based speaker adaptation. In *Proceedings of the ICASSP*, Salt Lake City, USA, 2001.

[8] P. Woodland and D. Povey. Large Scale Discriminative Training of Hidden Markov Models forh Speech Recognition. *Computer Speech and Language*, 6, 2002.

[9] Hua Yu and Alex Waibel. Flexible Parameter Tying for Conversational Speech Recognition. In *Proceedings of SSPR*, Tokyo, Japan, 2003.