Recent Experiments with HTK Broadcast News and Conversational Telephone Systems

Phil Woodland, Gunnar Evermann, Mark Gales, Thomas Hain,
Ricky Chan, Bin Jia, Do Yeong Kim, Andrew Liu,
David Mrva, Dan Povey, Khe Chai Sim, Marcus Tomalin,
Sue Tranter, Lan Wang & Kai Yu

January 22nd 2003

Cambridge University Engineering Department

What we've been doing on STT

- Rebuilding infrastructure for BN and CTS
- Experiments on applying techniques developed on CTS to BN
- Concentrated on simple systems here:
  - normally single pass decoding without adaptation
  - also 2002 CTS 10xRT system structure used
- Some new techniques including:
  - MPE/MMI-MAP
  - discriminative SAT
  - variable number of Gaussians/state
- first experiments with Mandarin
- evaluation of fast transcription

Background: 1998 BN & 2002 CTS eval systems

- All systems use PLP + first/second differentials
- CMN, CVN, VTLN used for normalisation
- Decision tree clustered cross-word triphones (full systems also quinphones)
- Eval systems operate in multiple passes and generate lattices and rescore with various adapted models and combine
- Eval systems use 4-gram LMs plus class-based trigram
- BN system used automatic segmentation: CTS system manual

Background: 1998 BN & 2002 CTS eval systems

- CTS system also includes
  - lattice-based MPE (minimum phone error) training
  - HLDA
  - lattice MLLR
  - SAT training
  - pronunciation probabilities
  - confusion networks for posterior decoding, confidence scores and system combination
- Fast, simplified versions (10xRT) of both BN an CTS systems exist

Corrected CTS transcriptions

A mistake in the Switchboard training transcriptions used in building all CUHTK CTS systems since 2000 was discovered.

- Training transcriptions are generated from MS-State text files using a number of filters which normalise the text and correct typos etc.
- During that process some words (`DID`, `THEY`, `IT`) were systematically deleted
- Overall about 3% of the word tokens were accidentally deleted
- Affected both acoustic models and LMs

The training transcriptions were regenerated using the final version of the MS-State transcriptions.

- Added 294 new conversation sides (total 4800 sw1 sides)
- Effectively 12h new data (new total 295.5 hours)

Corrected CTS transcriptions - Effects

All acoustic models were rebuilt from scratch with the new reference and tested with the old trigram LM.

	WER	
	old	new
MLE (dev01sub)	36.7	36.4
MLE+HLDA (dev01sub)	34.3	33.8
MPE+HLDA (dev01)	30.4	30.1
10xRT sys (eval02)	27.2	26.7

%WER on dev01(sub) and eval02, respectively. Using the same (old) trigram LM

- Small but consistent improvement of about 0.5% abs
- Model building process is surprisingly robust to transcription errors
- Still need to rebuild all LMs with new transcriptions

Rapid Transcription Experiments

Preliminary experiments to judge impact of rapid transcriptions vs. careful transcriptions

- Rapid Transcriptions for small amount of CTS made available a few days ago
- 185 conversation (370 conversation sides) from Switchboard-I
- Baseline: MSU careful, accurate transcriptions
  - 21.7 hours of data used in training
  - 8 mixture comps, 2529 triphone states
- Rapid Transcription: 5-10xRT transcription from LDC
  - discarded portions of data with *any* problems
  - 18.3 hours of data used
  - Rapid Transcription Model - 8 mixture comps, 2480 triphone states
- Hub5 Full Training set (296h) Model - 16 mixture comps, 6189 triphone states

Experimental Results

- Results on dev01 with trigram LM

	Overall	Swbd1	Swbd2	Cellular
Baseline MLE	43.4	33.6	47.9	48.9
RapidTran MLE	45.2	35.0	50.0	50.8
Baseline MPE	40.6	30.4	45.3	46.3
RapidTran MPE	41.2	30.9	45.9	47.1
Hub5FullTrain MLE	36.3	26.8	41.4	40.9
Hub5FullTrain MPE	32.0	22.4	37.0	36.9

- 1.8% error increase in MLE when we use rapid transcription (preliminary results: unresolved issues with segmentation)

- Only 0.6 % absolute error increase after applying MPE

- MPE robust to transcription quality

Discriminative MAP: Aims

- Perform MAP-style update without losing advantage of discriminative training
- E.g. MMI-MAP for MMI, MPE-MAP for MPE, see [Povey, Gales, Woodland: ICASSP2003]
- Requires more "adaptation" data than typical speaker adaptation, e.g. several hours. Applications include:
  - Task adaptation
  - Adaptation to a gender dependent system
  - Data-type specific models

Discriminative MAP: I-smoothing

- For robust discriminative training, esp. MPE, we combine objective function with a prior distribution over means & variances

- Center of prior is at the ML parameter estimates

- This is called I-smoothing– essential for good performance in MPE training

- Log prior distribution is $Q(\tau^I, \tau^I \mu_{jm}^{\text{prior}}, \tau^I ({\sigma_{jm}^{\text{prior}}}^2 + {\mu_{jm}^{\text{prior}}}^2) | \mu_{jm}, \sigma_{jm}^2)$, where $Q(\ldots)$ is the expected log-likelihood of τ^I points of data with mean μ_{jm}^{prior} and variance ${\sigma_{jm}^{\text{prior}}}^2$, given Gaussian parameters μ,σ^2

- The constant τ^I controls narrowness of prior distribution (as in normal MAP)

Discriminative MAP: priors (I)

- When discriminative MAP adaptation is performed, what should be the prior?

  i Discriminatively trained unadapted parameters, or
  ii ML-trained adapted parameters

- Sometimes i) may be better, sometimes ii), sometimes they may be similar

- Seems to be best to use ML-adapted parameters, even when improvement from MMI/MPE > normal improvement from ML-MAP

- May not be enough data to estimate adapted-ML parameters → ML-MAP for these

Discriminative MAP: priors (II)

- Need to estimate prior parameters $\mu_{jm}^{\text{prior}}, \sigma_{jm}^{\text{prior}}$

- μ_{jm}^{prior} and $\sigma_{jm}^{\text{prior}}$ are estimated with ML-MAP, so
  $\mu_{jm}^{\text{prior}} = \frac{\tau^{\text{MAP}}\mu_{jm}^{\text{orig}}+\theta_{jm}^{\text{mle}}(\mathcal{O})}{\tau^{\text{MAP}}+\gamma_{jm}^{\text{mle}}}$ and ${\sigma_{jm}^{\text{prior}}}^2 = \frac{\tau^{\text{MAP}}({\mu_{jm}^{\text{orig}}}^2+{\sigma_{jm}^{\text{orig}}}^2)+\theta_{jm}^{\text{mle}}(\mathcal{O}^2)}{\tau^{\text{MAP}}+\gamma_{jm}^{\text{mle}}} - {\mu_{jm}^{\text{prior}}}^2$

- τ^{MAP} is a constant set to e.g. 10 or 20, as in ML estimation

Discriminative MAP: priors (III)

- Gaussian update with prior is $\mu_{jm} = \frac{\theta^{\text{num}}_{jm}(\mathcal{O}) - \theta^{\text{den}}_{jm}(\mathcal{O}) + D_{jm}\mu'_{jm} + \tau^{I}\mu^{\text{prior}}_{jm}}{\gamma^{\text{num}}_{jm} - \gamma^{\text{den}}_{jm} + D_{jm} + \tau^{I}}$

  and $\sigma^2_{jm} = \frac{\theta^{\text{num}}_{jm}(\mathcal{O}^2) - \theta^{\text{den}}_{jm}(\mathcal{O}^2) + D_{jm}({\mu'}^2_{jm} + {\sigma'}^2_{jm}) + \tau^{I}({\mu^{\text{prior}}_{jm}}^2 + {\sigma^{\text{prior}}_{jm}}^2)}{\gamma^{\text{num}}_{jm} - \gamma^{\text{den}}_{jm} + D_{jm} + \tau^{I}} - \mu^2_{jm}$

- Equivalent to adding $\frac{\tau^{\text{MAP}}}{\tau^{\text{MAP}} + \gamma^{\text{mle}}_{jm}}\tau^{I}$ points of data with mean & var equal to unadapted Gaussian parameters, and

- $\frac{\gamma^{\text{mle}}_{jm}}{\tau^{\text{MAP}} + \gamma^{\text{mle}}_{jm}}\tau^{I}$ points of data with mean & variance equal to ML estimate from adaptation data

Discriminative MAP for task adaptation

- Adapting Switchboard system to Voicemail as example (later look at GD modelling for BN)

- Used with MMI (so MMI-MAP)

- Start from MMI-trained Switchboard system

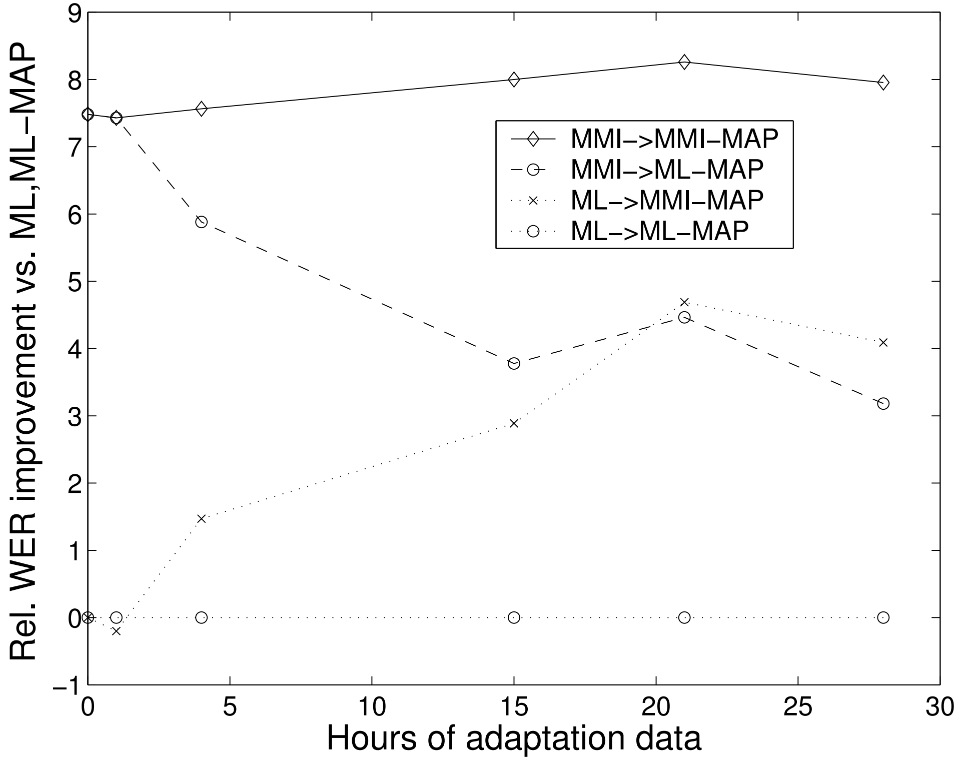
- WER reduced from 47.0% (unadapted) to 40.5% with MMI-MAP with 20 hours voicemail data

- (ML-)MAP adaptation gave 42.6% WER

- MMI-MAP gives 4.6% relative improvement compared to ML-MAP

- Importantly MMI-MAP retains the full benefits of discriminative training when performing adaptation (cf. applying (ML-)MAP).

MMI-MAP: Switchboard to Voicemail



MMI-MAP retains improvement of MMI over ML

Overview of BN Experiments

- BN System Setup
- Acoustic Training/Test Data
- Experiments on
  - HLDA
  - Gender-dependent modeling
  - MMI & MPE Training
  - MPE+HLDA
  - MPE-MAP

BN English System Setup

- Front-end
  - 12 MF-PLP cepstral parameters + C0 and 1st/2nd derivatives + segment CMN (no VTLN or CVN)
  - optional 3rd derivatives + HLDA
- Automatic segmentation into wideband/narrowband and gender
- Acoustic modelling
  - Decision tree state clustered, context dependent triphone models (6976 clustered states, 16-component mixture Gaussian)
  - Gender-dependent & band-dependent acoustic modelling
  - MMI/MPE/MPE-MAP training
- Trigram language model
- Single pass decoding (no adaptation!)

Acoustic Training/Test Data

Train on the combined set of 1997 and 1998 data

- **1997 data** 72 hours of acoustic BN training data
- **1998 data** 71 hours of acoustic BN training data

Development test sets

BNeval98 two 1.5-hour data sets

BNeval02 1-hour data set

HLDA

- Estimate HLDA transform based on MLE baseline system

	Total	F0	F1	F2	F3	F4	F5	FX	F	M
BNeval98										
non-HLDA	19.6	11.1	20.1	25.8	20.9	20.1	28.1	35.0	20.0	18.7
HLDA	17.9	10.2	18.5	22.6	19.1	18.9	27.2	30.5	18.2	17.1
BNeval02										
non-HLDA	17.9	–	–	–	–	–	–	–	20.1	17.0
HLDA	16.0	–	–	–	–	–	–	–	18.4	15.1

%WER of HLDA system on BNeval98 and BNeval02

- Add 3rd derivatives + HLDA, project 52 dim to 39

- Consistent improvement over different evaluation sets and F-conditions

Gender-dependent modelling

- Perform ML-update of means anx mix weights on gender-specific training subsets

	Total	F	M
non-HLDA			
GI	19.6	20.0	18.7
GD	18.8	18.1	18.7
HLDA			
GI	17.9	18.2	17.1
GD	17.1	16.4	17.0

%WER for GI & GD on BNeval98

- Most gains come from female speakers (2/3 of training data male)
- Gains still present with HLDA

MMI & MPE Training

	Total	F0	F1	F2	F3	F4	F5	FX	F	M
BNeval98										
MLE	19.6	11.1	20.1	25.8	20.9	20.1	28.1	35.0	20.0	18.7
MMI	17.0	9.9	17.5	24.1	18.7	17.1	20.4	29.4	16.6	16.6
MPE	16.2	9.6	17.1	22.6	17.5	16.1	21.7	27.8	16.2	15.6
BNeval02										
MLE	17.9	–	–	–	–	–	–	–	20.1	17.0
MPE	15.0	–	–	–	–	–	–	–	16.7	14.3

%WER on BNeval98 and BNeval02.

- MMI and MPE reduce WER by 2.6% and 3.4% absolute, respectively
- Both are most effective on difficult data (FX)
- MPE outperforms MMI on BN

MPE+HLDA

	Total	F0	F1	F2	F3	F4	F5	FX	F	M
BNeval98										
MPE	16.2	9.6	17.1	22.6	17.5	16.1	21.7	27.8	16.2	15.6
MPE+HLDA(1)*	15.0	8.7	15.8	20.2	17.5	15.4	19.1	25.3	15.1	14.4
MPE+HLDA(2)	15.0	8.8	15.5	19.6	17.3	15.3	19.1	25.7	15.1	14.3
BNeval02										
MPE	15.0	–	–	–	–	–	–	–	16.7	14.3
MPE+HLDA(1)	14.0	–	–	–	–	–	–	–	16.0	13.3
MPE+HLDA(2)	13.6	–	–	–	–	–	–	–	15.6	12.8

%WER on BNeval98 and BNeval02. MPE+HLDA(1) was trained based on lattices generated with non-HLDA while MPE+HLDA(2) used HLDA lattices
(* This system was used for dry-run.)

- MPE+HLDA outperforms MPE in all F-conditions
- HLDA lattices better on BNeval02, same on BNeval98

MPE-MAP with HLDA for GD modelling

iter	Total	F	M
0	15.0	15.1	14.4
1	14.7	14.4	14.3
2	14.5	14.0	14.3
3	14.5	14.0	14.3
GD*	14.8	14.4	14.5

%WER for MPE-MAP with HLDA on BNeval98

* GD denotes GD-MPE with HLDA.

- MPE-MAP GD reduces WER by 0.5% absolute over MPE-MAP
- Better than GD MPE by 0.3% absolute (1 iteration on split data set)
- Most gains come from female speakers while both genders were improved

BN-E Experiments: Conclusions

- Demonstrated the utility of MPE and HLDA on BN data
- Shown that MPE-MAP can be used to produce effective GD systems
- Forthcoming...
  - Lattice MLLR + Full-Variance transform adaptation
  - Interpolated 4-gram + class-based LM
  - Posterior probability decoding via confusion networks
  - SAT training?
  - System combination?
  - Vocal tract length normalisation?
  - Quinphone model (not for 10xRT system)

Discriminative Speaker Adaptive Training (DSAT)

- Speaker Adaptive Training (SAT) constructs canonical models: includes speaker-specific linear transforms into parameter re-estimation.

- Constrained (same transform for mean and variance) model-based transformation can be applied to the feature space: makes model re-estimation more straightforward.

- In RT-02 CTS SAT training
  - used constrained MLLR for each conversation sides
  - interleaved re-estimation of HMM parameters using MPE and MLLR transforms

- Here, apply discriminative criteria for constrained linear transforms and then re-estimate HMM parameters with a consistent objective function.

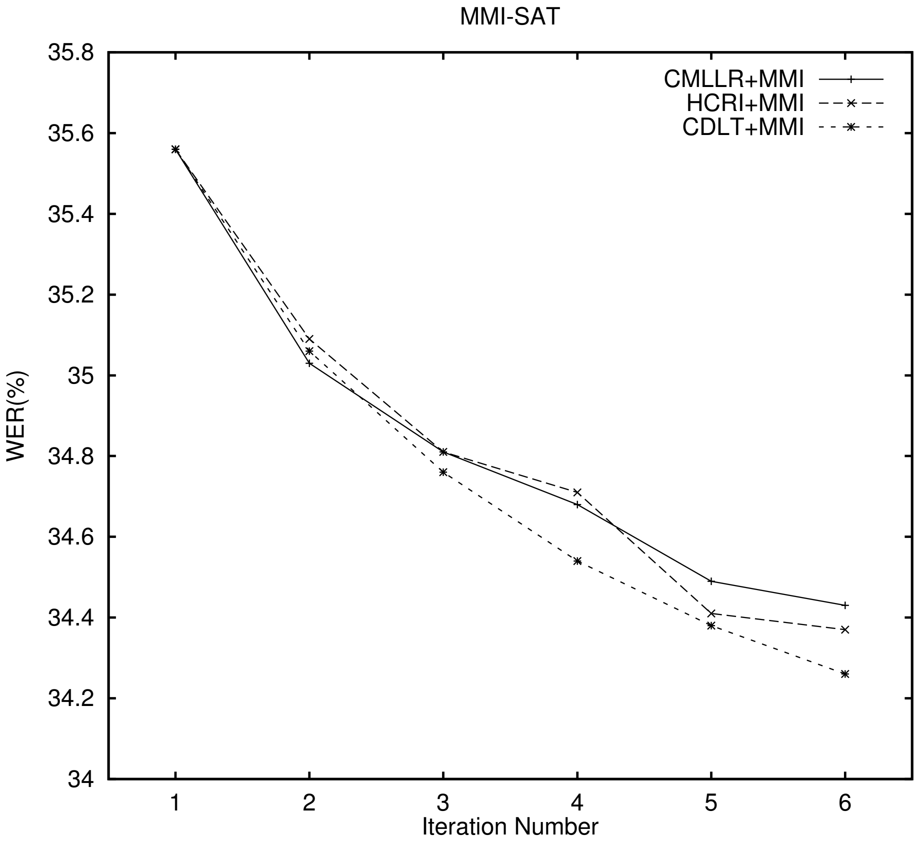
- Constrained transform estimation can use H-criterion (interpolation of ML and MMI) [Uebel & Woodland, 2001]

- Constrained Discriminative Linear Transformation (CDLT) uses transform with a pure MMI objective function [Byrne et al, 2002].

- Initial experiments used Switchboard minitrain, and mttest is used for test-set.
  - Starting with normal SI models, 4 iterations of interleaved transform estimation and model parameter updating are performed to obtain ML-SAT models.
  - 5 (or 8) iterations of interleaved transform estimation and model parameter updating yield MMI-SAT models.
  - Test-set adaptation still uses MLLR adaptation

- In future, investigating use of MPE criterion for linear transform generation.

DSAT Preliminary Results



%WER for different types of adaptive training

Controlling #Gaussians per state

- Standard approach to setting $\#$Gauss was to have N Gaussians per speech state and $2N$ for silence
- E.g. for N in the range 12 to 30 depending on the system.
- Investigated setting $\#$Gauss as a function of number of frames γ_j available to train state j
- Use $\#\text{Gauss} = k\gamma_j^p$, where p is a small power (e.g. 1/5)
- k is a normalising constant set to make the average $\#$Gauss equal to N

Controlling #Gaussians in states: experiments

- Preliminary experiments found that a wide range of powers p can work
- $p = 1/5$ is generally a suitable value
- Use same average $\#$Gauss/state N as baseline
- Gives consistent improvements:

Setup	Test	fixed N	variable N
CTS MLE, N=28	dev01	33.8	33.5
CTS MPE, N=28	dev01	30.2	29.9
BN MLE, N=16	bneval98	17.9	17.6

Controlling #Gaussians in states: distribution

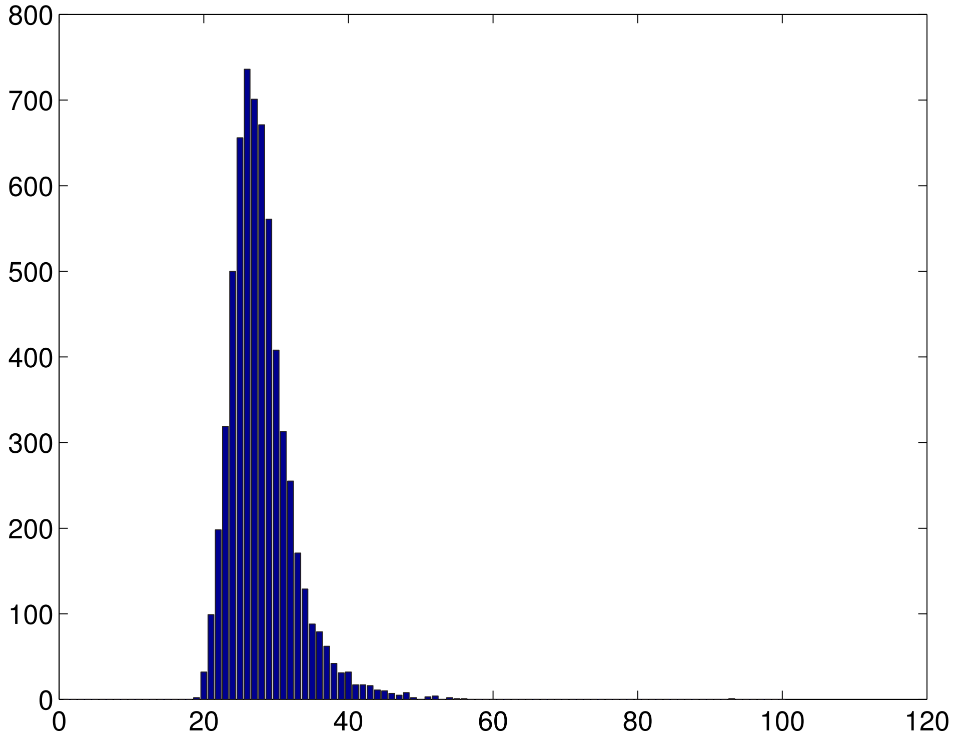


Figure 1: Distribution of #Gauss among states

- All states have #Gauss in range 19-56, except silence (around 100 Gauss/state)

Mandarin CTS System Overview

- Same front end as English CTS: PLP, CMN, CVN + VTLN
- Decision tree state clustered, cross-word triphones
- Gender independent, 60 tone-independent phones
- 11.6k words dictionary (subset of 44k LDC dictionary + 1.5k English words)
- Acoustic Training: 17 hours CallHome + 20 hours CallFriend
- Test (eval01): 2 hours Hub5 Mandarin 2001 evaluation data (20 conversations)
- LM used data segmented into words using LDC segmenter with text normalisation (numbers & digits)
- 578k words acoustic training and 9918k words newspaper texts
- Interpolated trigram LM from acoustic (0.91) & newspaper (0.09)

Initial Mandarin Results

LM Perplexity Test

LM	Bigram	Trigram
acoustic	255	259
newspaper	2109	1950
interpolation	237	238

Perplexity tested on eval01

System Performance

System	%CER
triphone (from flat-start)	61.5
+ 2-model re-estimation	60.1
+ test VTLN	59.7
+ train&test VTLN	58.4

%CER for the baseline Mandarin systems on eval01

Automatic Segmentation for CTS

- A simple GMM based segmenter with gender-dependent, channel-specific (landline vs. cellular) models was built (details in MDE talk)

Preliminary experiments with 10xRT CUHTK system (3 passes):

- Full eval02 (6h) with “old” scoring (`sclite` and RT02 STMs)
- Manual PEM segmentation provided for RT02 used as baseline
- Automatic segmentation constrained by outer PEM boundaries
- Optionally resegment using first pass transcription (removing long silences)

	Sub	Del	Ins	WER
PEM	17.1	6.1	3.4	26.7
seg1	17.0	6.9	4.2	28.1
seg1-r	16.9	7.0	4.0	28.0

Automatic Segmentation for CTS – new scoring

- Segment whole audio file
- dev03 subset (1h) with "new" scoring (`stt-eval` and new ref CTM with $t_{pad} = 2.0s$ to compensate for "George's New Math")
- **ctmseg** generated from reference CTM (0.6s silence smoothing, 0.2s padding)
- **seg1** uses 0.6s silence smoothing, **seg2** uses 1.2s (worse for Diarisation)

	Sub	Del	Ins	WER
PEM*	18.28	7.37	3.77	29.43
ctmseg	18.56	6.07	3.67	28.30
mitbase	18.46	7.71	3.83	30.00
seg1	18.58	6.59	3.93	29.10
seg1-r	18.49	6.92	3.77	29.18
seg2	18.37	6.63	4.00	28.99

*submitted for the dry run

Automatic Segmentation – Conclusions & Future Work

- Automatic segmentation on CTS costs about 1% abs.
- Segmentation needs to be tuned specifically for STT
- Build more complex speech/silence models
- Study effects of silence smoothing and padding on channel normalisation
- Investigate effect of different segment lengths on LM effectiveness
- Make better use of word times from early transcription passes

HTK Development – Aims

HTK Development is one of the three EARS tasks at CUED

The aims of this task are:

- Make state-of-the-art LVCSR technology available
- Infrastructure for research in academia and industry, helps smaller groups
- Allow researcher to focus on one part of the problem (e.g. LM) but still test in state-of-the-art system
- HTK provide full source code of implementations and extensive documentation
- Tool for teaching
- Document research results and provide vehicle for technology-transfer
- Provide baselines for a range of tasks and testsets

HTK Development – Background

- The HTK (Hidden Markov Model Toolkit) has been at the core of ASR research at Cambridge for more than ten years
- Very flexible, modular toolkit written in ANSI C
- HTK3: available for free download since Sep 2000

  `http://htk.eng.cam.ac.uk`

  More than 16,000 registered users, heavily used in teaching, active support mailing lists
- Important features for state-of-the-art ASR systems are missing
- In the past there was a big difference between public HTK and the CUED-internal "CU-HTK"

HTK Development – Progress in EARS

- Internally at CUED: grand code unification. All EARS-related work now uses a single unified, version-controlled source tree.
- Snapshots of (large parts of) this source tree are regularly released as public HTK3 versions
- First release from this tree was HTK3.2, December '02, new features include:
  - HLM language modelling toolkit
  - support for global feature space transforms (e.g. used for HLDA)
  - Lattice processing tool allows lattice pruning, search and expansion
  - *2-model re-estimation* (important for state clustering)

HTK Development – Progress in EARS

- HTK-users meeting at ICASSP 2002 in Orlando
- Complete re-implementation of flexible transform-based adaptation framework
- Large Vocabulary Decoder
- Re-estimation tools for discriminative training integrated
- Initial “How to build a BN system” recipe created

HTK Development – HLM

- Comprehensive set of tools for training back-off n-gram language models
- Scales well to very large text corpora
- Supports word- and class-based LMs
- Includes clusterer to automatically find classes based on bigram statistics
- Supports linear interpolation and merging of LMs

HTK Development – LVR decoder

- Time-synchronous static single lexicon tree decoder
- Uses HTK cross-word triphones
- Supports arbitrary finite-state LMs
- Allows lattice generation and rescoring
- Can generate phone-marked lattices for lattice-based adaptation or training
- Geared to be flexible and not tuned for speed like some of our other decoders
- Has been used in Hub5 RT02 for lattice generation and rescoring

HTK Development – new adaptation

Adaptation for HTK3.3 has been extended:

- Highly flexible extensible framework:
  - integrated with HTK style macros - allows parameter/transform tying
  - support for complex transforms e.g. CAT, cascade transforms etc.
- Multiple types of transformation - currently implemented:
  - mean MLLR adaptation (`MLLRMEAN`)
  - full-variance MLLR adaptation (`MLLRCOV`)
  - constrained MLLR adaptation (`CMLLR`)
- Framework also supports:
  - lattice-based estimation
  - hierarchies of transforms
  - adaptive training
  - ability to select transform for alignment and “parent”
  - transform interpolation

Example Hierarchy of Transforms

```
~a “mjfg”
<ParentXForm> ~a “cued”
<AdaptKind> TREE
<BaseClasses> ~b “regtree.base”
<XFormSet>
    <XFormKind> MLLRMEAN
    <NumXForms> 1
    <LinXForm> 1 <VecSize> 3
       <OffSet>
       <Bias> 3
          -0.357 0.001 -0.002
       <BlockInfo> 1 3
       <Block> 1
          <XForm> 3 3
             0.942 -0.032 -0.001
             -0.102 0.922 -0.015
             -0.016 0.045 0.910
<XFormWgtSet>
    <ClassXForm> 1 1
    <ClassXForm> 2 1
```

```
~a “cued”
<AdaptKind> CLASS
<BaseClasses> ~b “global”
<XFormSet>
    <XFormKind> MLLRCOV
    <NumXForms> 1
    <LinXForm> 1 <VecSize> 3
       <LogDet> -0.0051
       <BlockInfo> 1 3
       <Block> 1
          <XForm> 3 3
             1.01 0.30 0.20
             0.10 0.97 0.20
             0.10 0.20 1.10
<XFormWgtSet>
    <ClassXForm> 1 1
```

HTK Development – discriminative training

- Re-estimation tools for lattice-based discriminative training
- Relies on LVR decoder to produce phone-marked lattices
- Support MMIE and MPE
- Supports HTK-style two pass training (parallel stats accumulation followed by actual re-estimation pass)
- Supports MMI-MAP and MPE-MAP

HTK Development – Plans & Conclusions

Plans for the next 12 months:

- Finalise WSJ recipe
- Document new adaptation framework with examples of advanced techniques, e.g. Cluster Adaptive Training
- Release HTK 3.3 (incl. new adaptation code)
- Document discriminative training & LVR decoder
- Public availability of HTK3 continues to be a success: more than 16,000 registered users, about 400 downloads of HTK-3.2 since Christmas.
- HTK-users meeting at ICASSP 2003 in Hong Kong