

The AMI RT'09 STT and SASTT Systems



Thomas Hain, Asmaa El Hannani, Vincent Wan - UoS
Lukas Burget, František Grézl, Martin Karafiat - BUT
John Dines, Phil Garner- IDIAP
Marijn Huijbregts - Univ. Twente
Peter Bell, Mike Lincoln - Univ Edinburgh

May 28, 2009

Outline

- ▶ Review of the RT'07 systems

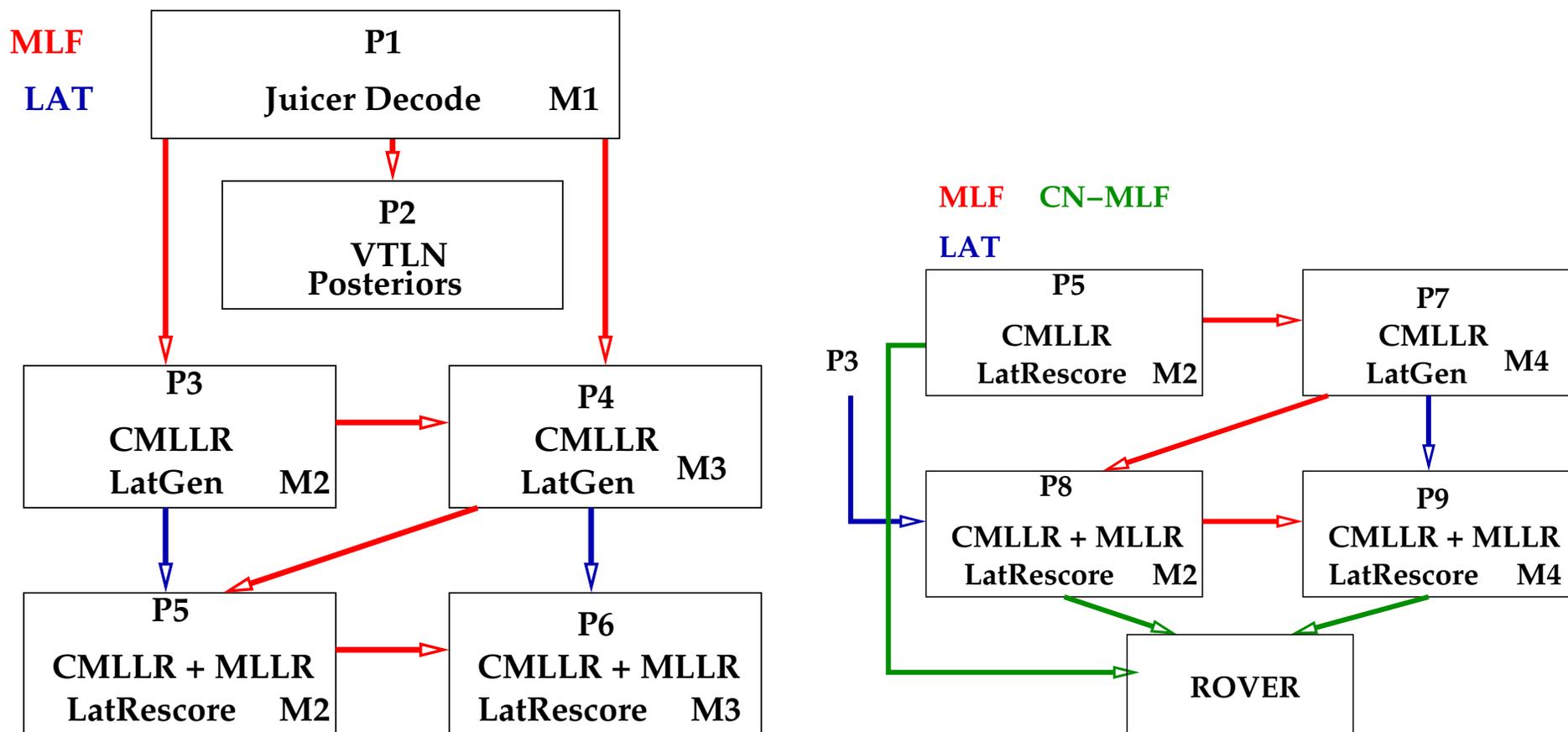
- ▶ Improvements and experiments for RT'09
 - ▷ Performance
 - ▷ Speed
 - ▷ Structure

- ▶ The RT'09 - STT results
 - ▷ Analysis
 - ▷ Bugfix systems

Review of the 2007 System - Key features

1. Dictionary and word list expansion and cleaning
2. New training data (and hence new models)
 - (a) *ihmtrain07* and *mdmtrain07*: includes new NIST and AMI data
 - (b) *ctstrain07*: now includes 2000 hours of Fisher data
3. NB/WB Adaptation
4. IHM segmentation optimisation
5. Included AMI MDM segmentation and clustering
6. Alternative front-end: MFCC + Bottleneck features
7. ROVER / CNC
8. System architecture

2007 System Architecture



2007 Performance Conference Meeting -IHM - *rt07seval*

	TOT	Sub	Del	Ins	CMU	EDI	NIST	VT
P1	37.4	20.6	12.9	4.0	41.5	28.4	18.8	41.3
P3.fg	28.2	14.5	10.4	3.3	33.7	19.8	14.1	30.8
P4	27.9	14.1	10.6	3.2	33.1	20.0	13.8	30.2
P5	27.7	13.5	11.1	3.1	34.5	19.5	13.6	30.4
P5.cn	25.9	13.5	9.9	2.5	31.2	18.3	12.0	28.5
P6.cn=final	25.7	13.6	9.5	2.6	30.6	18.4	11.8	28.2
P7	27.9	14.5	9.9	3.4	34.7	20.3	13.9	29.6
P8	26.9	13.6	10.1	3.3	32.0	19.4	13.3	29.6
P8.cn	25.4	13.4	9.4	2.6	30.8	18.0	11.7	27.2
P9	27.9	14.6	9.9	3.5	34.7	20.4	14.0	29.6
P9.cn	26.3	14.3	9.3	2.7	33.5	19.0	12.3	27.1
P5+P8+P9	24.9	12.7	9.8	2.4	30.5	17.6	11.5	26.8

2007 Performance Conference Meeting - *rt07seval* - MDM

	ICSI S&C				AMI/DA S&C			
	TOT	Sub	Del	Ins	TOT	Sub	Del	Ins
P1	44.2	25.6	14.9	3.8	44.7	25.7	16.3	2.7
P3	38.9	18.5	16.8	3.5	34.5	19.3	12.5	2.7
FINAL	33.7	20.1	10.7	2.9	33.8	19.2	12.2	2.4
FINAL manual seg	30.2	18.7	9.4	2.0	-	-	-	-

- ▶ Substantial differences between segment's
 - ▷ Performance level may hide weaknesses

- ▶ Manual segmentation substantially better

New in the 2009 System

- ▶ Modelling
 - ▷ IHM Segmentation - New training data
 - ▷ Beamforming - Using ICSI BeamformIt
 - ▷ Speaker clustering - Using the AMI diarisation system
 - ▷ Stacked Bottleneck features
 - ▷ fMPE
 - ▷ Full meeting adaptation
 - ▷ Language modelling and wordlist - slight modifications
- ▶ Infrastructure
 - ▷ Juicer updates - speed and performance
 - ▷ ROTK
- ▶ Data
 - ▷ New AMIDA data
 - ▷ Data selection

Overall our systems got simpler and much faster

Not quite made it

1. CTS training
2. Full covariance modelling
3. Discriminative adaptation
4. Automatic system optimisation
5. ...

Data

- ▶ Only using meeting data this year !
- ▶ Inclusion of AMIDA training data
 - ▷ Approx. 8 hours of data
 - ▷ Similar to RT'09 EDI data
 - ▷ At the same time, re-segmentation of the complete training set with 100ms collar: 191 hours total
- ▶ IHM performance
 - ▷ ML baseline identical but higher number of deletions triggering issues with scale factors
 - ▷ After VTLN degradation by 0.3% WER absolute
- ▶ MDM performance
 - ▷ Again slight degradation in performance

MDM: Automatic data selection

► Experiments testing training style

- ▷ multi-channel training gave poorer performance
- ▷ Overlap segmentation at close word boundaries gave best results
- ▷ 154 hours of data (129 before !)

► Cleaning of data using confidence scores

- ▷ Confidence of word transcription in generated lattice (bigram)
- ▷ Delete utterances based on C_{max} : Highest posterior in lattice

Data size	TOT	Sub	Del	Ins
80% ML	42.6	26.7	13.4	2.5
80% MPE	40.7	24.1	14.4	2.2
90% ML	42.2	26.2	13.5	2.5
90% MPE	40.5	24.0	14.4	2.1
95% ML	42.8	26.8	13.5	2.5
95% MPE	40.7	23.3	15.4	2.1
100% ML	42.8	26.8	13.6	2.5
100% MPE	40.8	23.5	15.3	2.0

Experiments on rt07seval MDM reference segmentation

Language modelling and word lists

- ▶ LMs trained using same corpora and tools as per RT07
 - ▷ No additional training corpora or web data collections were used
 - ▷ Main change:
 - Cut-offs were lowered slightly. e.g. min count of 4-gram counts was set to 3 instead of 4 as in previous evaluations.
 - Larger but better language models.

PPL	RT07 4g LM	RT09 4g LM
RT07s ihm	87.8	86.4
RT09s ihm	73.1	71.0

rt07seval	RT07 4g	RT09 4g
IHM	37.9	36.7
MDM	36.2	35.9

%WER using pruned LMs and HDecode.

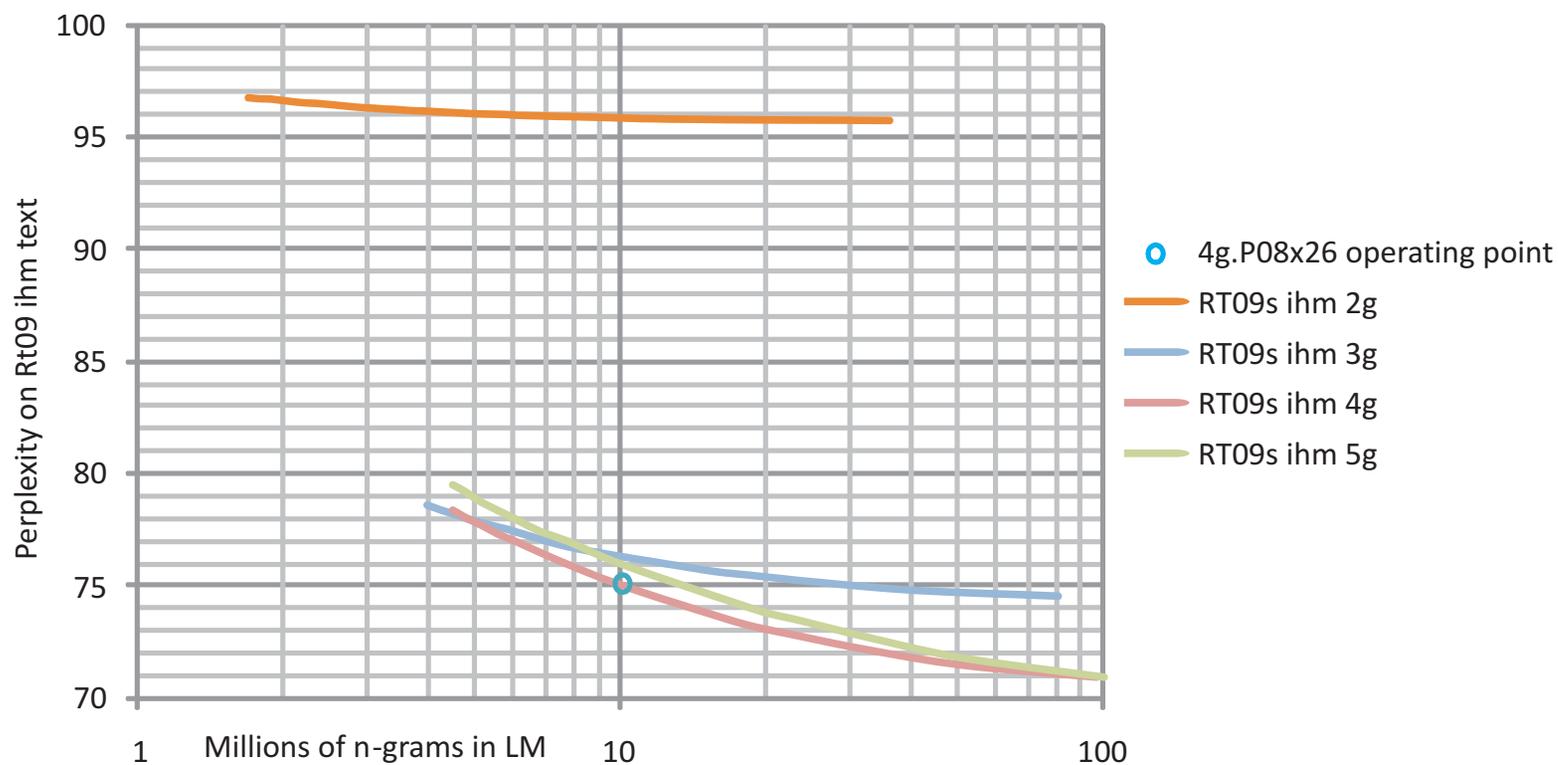
OOV%	RT07 4g LM	RT09 4g LM
RT07s ihm	0.74%	0.62%
RT09s ihm	0.30%	0.29%

All results use unpruned LMs

Language model pruning

► Pruning became important due to use of WFST based decoding

▷ SRI LM toolkit entropy pruning



Perplexities on RT09 IHM reference

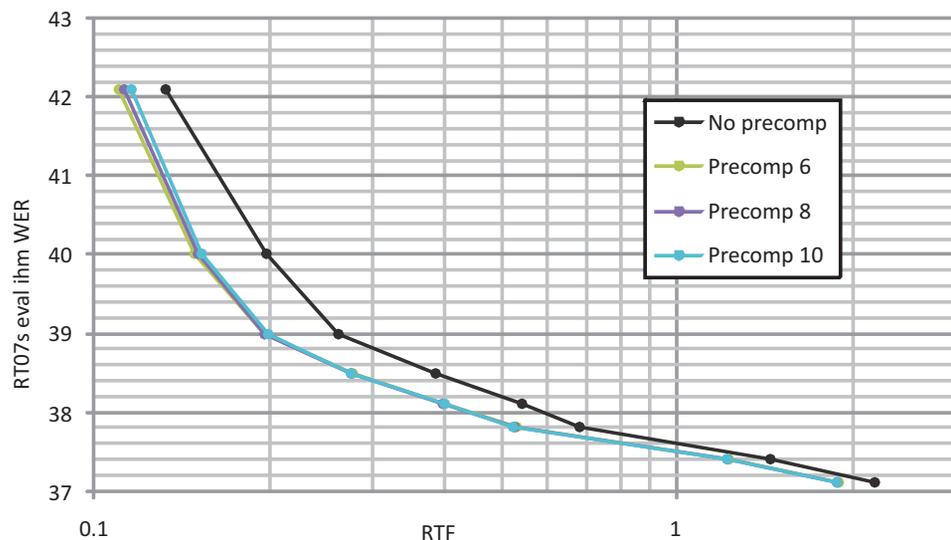
RT09 IHM - Perplexities with pruned LMs

- ▶ RT09 data was not difficult in terms of LMs

	EDI	IDI	NIST	Combined
OOVs	17	48	49	115
Words Sent	10980	12292	15368	38640
4g pruned PPL	72.1	86.1	70.5	75.5

- ▶ The data set with the highest error rate shows the lowest perplexity.

Juicer: Improvements



- ▶ Rewrite of token-passing core
- ▶ Optimised juicer internal structures to use the L2 cache more efficiently
- ▶ Use optimised Intel libs for GMM calculation

- ▶ Precalculation of GMM (graph)
- ▶ Tight integration with HTKLib. Anything HTK can do juicer can do too
- ▶ LM factors not encoded in network
- ▶ HMM topology not encoded in network

Juicer: Speed as a function of WFST size

1. Entropy pruning and decoding using fixed beam settings

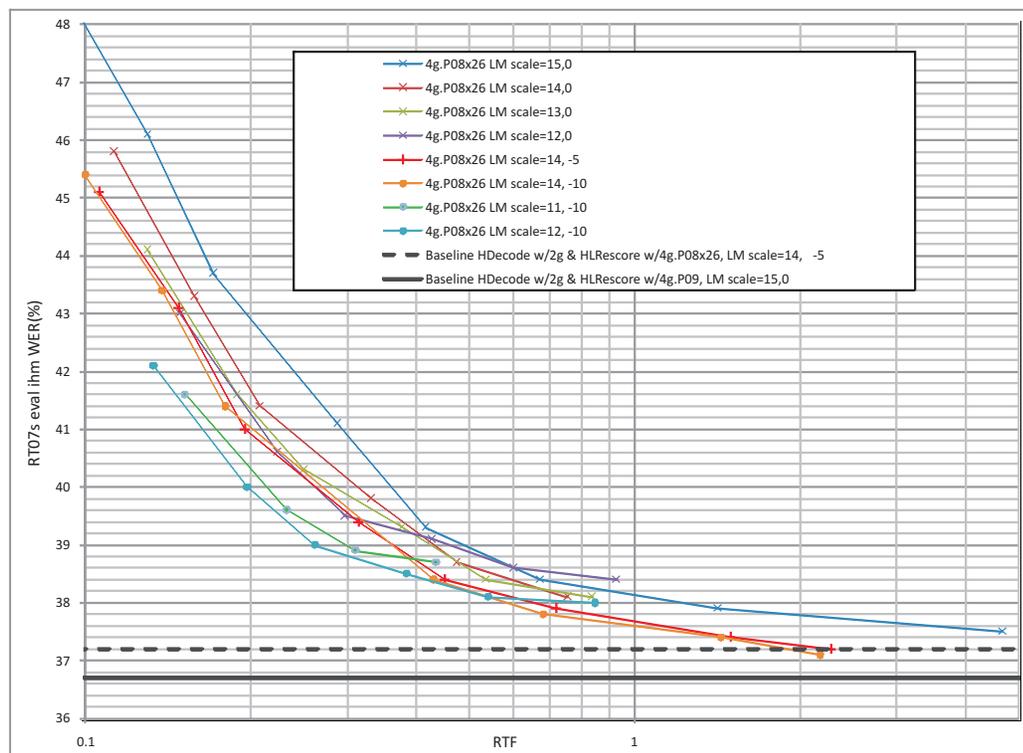
Total n-grams in 4g LM	Arcs in WFST	WER	RTF
3.5M	15.6M	46.8	0.579
4.4M	19.5M	46.5	0.591
6.1M	26.6M	46.6	0.597
8.0M	35.2M	46.7	0.606

2. Change of lexicon size and LM order

Lexicon size	LM order	Arcs in WFST	WER	RTF
2K	7	11.8M	55.3	0.827
6K	7	12.5M	48.2	0.625
10K	7	13.8M	47.2	0.582
16K	7	14.7M	46.8	0.589
50K	4	15.6M	46.8	0.579

Build the biggest LM using the minimal OOV for any given speed !

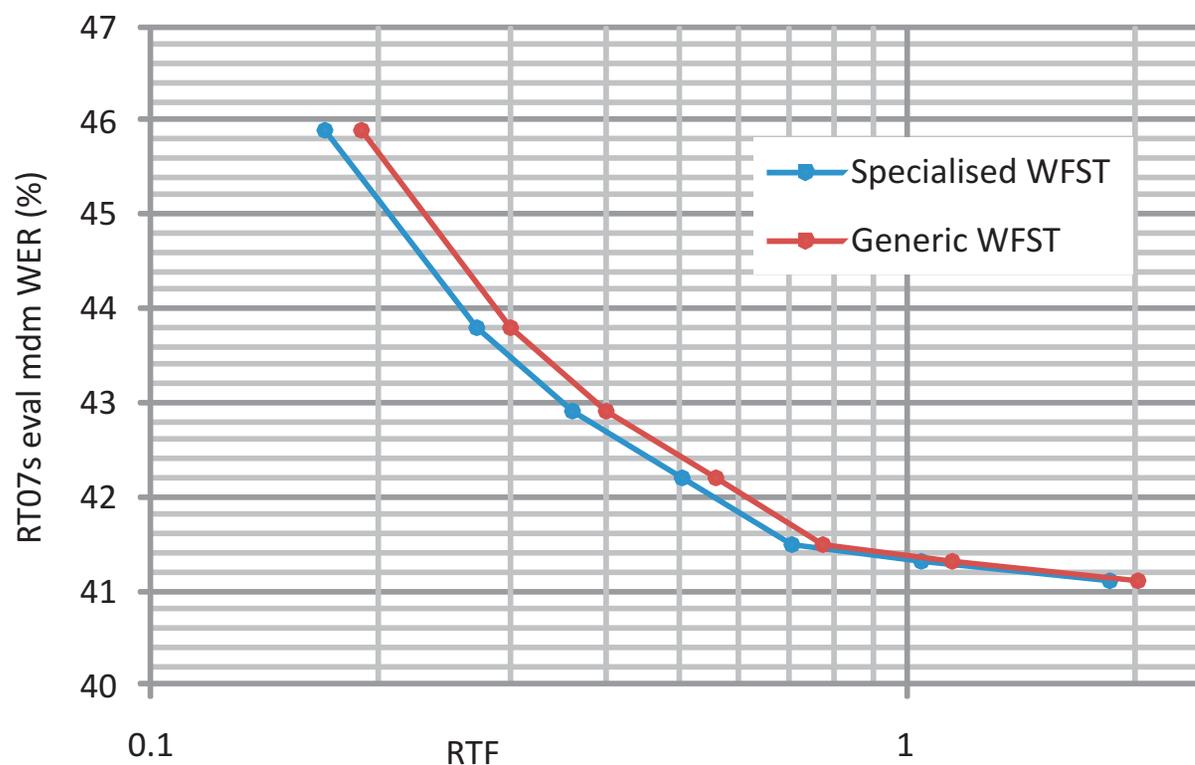
Juicer: WER vs RTF for various LM scale factors



1. Juicer performs as well as HDecode given the same LM
2. Juicer is faster than HDecode + HLRescore achieving the same accuracy at just over 2xRTF
3. Fast decoding optimisation requires changes to LM factors

Juicer: Untying of states - Speed

- ▶ Convenient for RT evals but leads to a slightly slower decode (about 10% slower).



IHM front-end: Speech Activity Detection

- ▶ Same basic setup as in previous years
 - ▷ Training: MLP based speech/silence classifier using MF-PLP + cross-talk features
 - ▷ Segmentation: Viterbi decoding of scaled likelihoods
 - ▷ Tuning: HMM minimum duration, speech/silence class priors, insertion penalty

- ▶ Differences from sys07 IHM segmentation
 - ▷ Training on sys09 training data set (more AMI(DA) data)
 - ▷ Tuned on RT07s conf. eval

IHM front-end: Development on *rt07seval*

- ▶ Reduction in WER 0.9% absolute over the AMI 2007 system
- ▶ Similar number of segments as reference

System	#Segments	%WER							
		Tot	Sub	Del	Ins	CMU	EDI	NIST	VT
ref	4527	29.3	18.9	7.7	2.7	36.7	24.5	24.5	31.2
auto-sys07	2717	32.6	17.7	10.9	4.0	41.2	26.2	29.1	33.3
auto-sys09	4541	31.7	18.1	9.5	4.0	42.4	25.3	26.8	31.7

- ▶ %WER shown for a simplified two-pass speaker adaptive ASR system

IHM front-end: Performance on *rt09eval*

- ▶ Good performance on EDI and IDI
- ▶ Horrible performance on NIS – far too many deletions!
- ▶ Class priors can be used to tune proportion of speech/silence

System	$P(sil)$	#Segs	WER %						
			Tot	Sub	Del	Ins	IDI	EDI	NIS
ref	–	5660	32.9	22.1	7.1	3.7	37.9	27.7	32.5
auto	0.80	4809	36.4	20.6	11.9	3.9	38.8	28.5	40.2
auto	0.85	4949	36.1	20.9	10.7	4.4	39.9	28.6	38.4
auto	0.90	5135	36.4	21.0	10.7	4.5	42.3	28.6	37.2
auto	0.95	5504	39.8	22.0	8.5	9.4	51.7	29.1	37.7

- ▶ *sys09* configuration is shown in **bold**, %WER shown for adapted RT09 system output

IHM front-end: Analysis on RT09s

- ▶ Oracle' systems – optimal choice of class priors w.r.t
 - ▷ Entire eval set (Oracle-Global)
 - ▷ Per meeting (Oracle-Meeting)
 - ▷ Per channel (Oracle-Participant)

- ▶ Tuning of class priors can give significant improvement – though doesn't completely solve our problems

- ▶ We don't yet know how to automatically choose the best class-priors

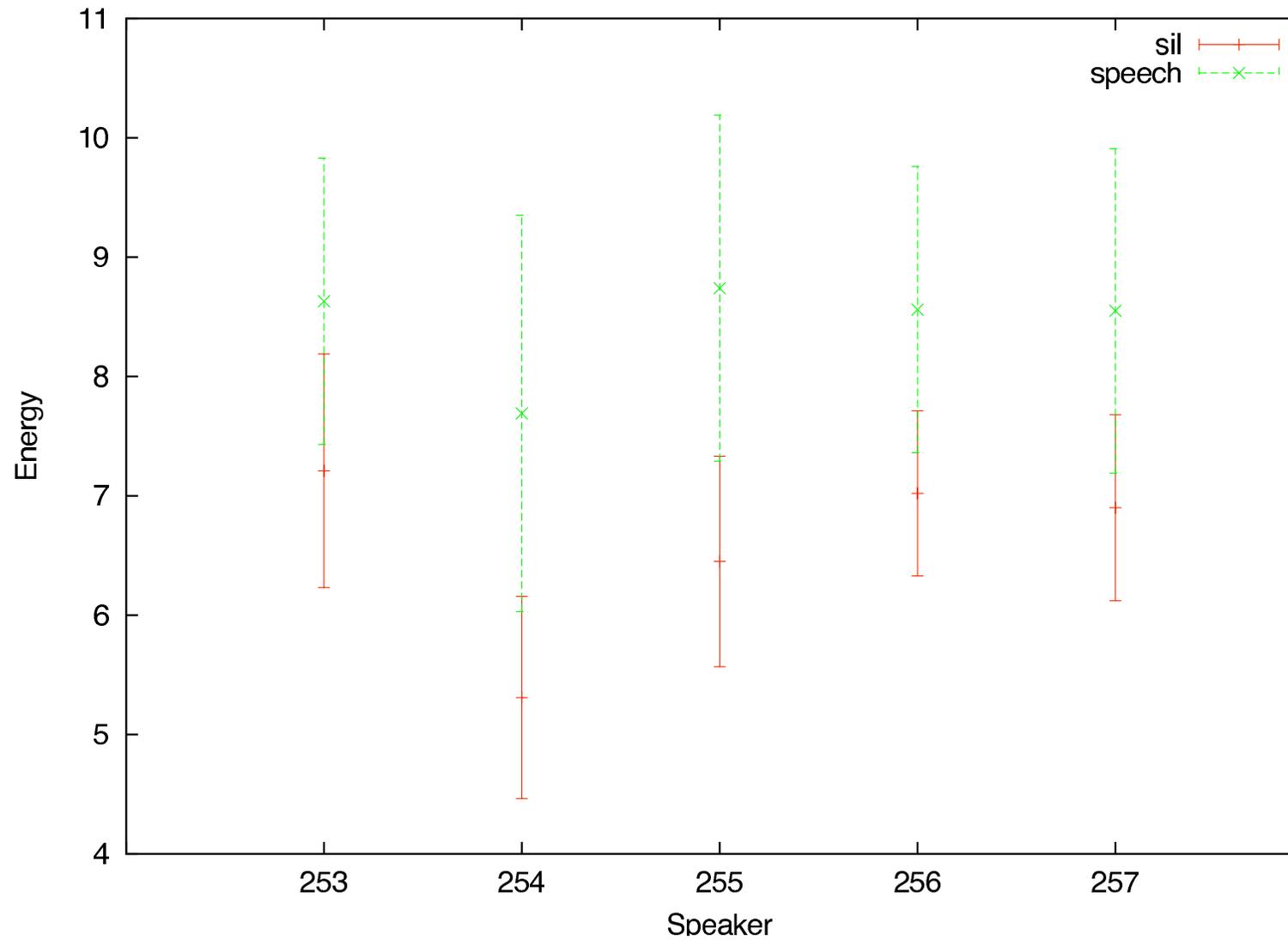
System	$P(sil)$	WER %						
		Tot	Sub	Del	Ins	IDI	EDI	NIS
auto	Oracle-Global	36.1	20.9	10.7	4.4	39.9	28.6	38.4
auto	Oracle-Meeting	35.1	21.3	9.0	4.9	38.8	28.4	36.8
auto	Oracle-Participant	34.9	21.2	8.9	4.8	38.9	28.3	36.5

IHM front-end: What else went wrong?

- ▶ Why was the NIST data so challenging?
 - ▷ Different recording levels across channels in the same meeting
 - ▷ Different relative levels of cross-talk across channels in the same meeting – different mics???

- ▶ We don't attempt to normalise features across channels within the same meeting
 - ▷ We didn't need to up until now – but it now looks like we should!

IHM front-end: Energy across channels



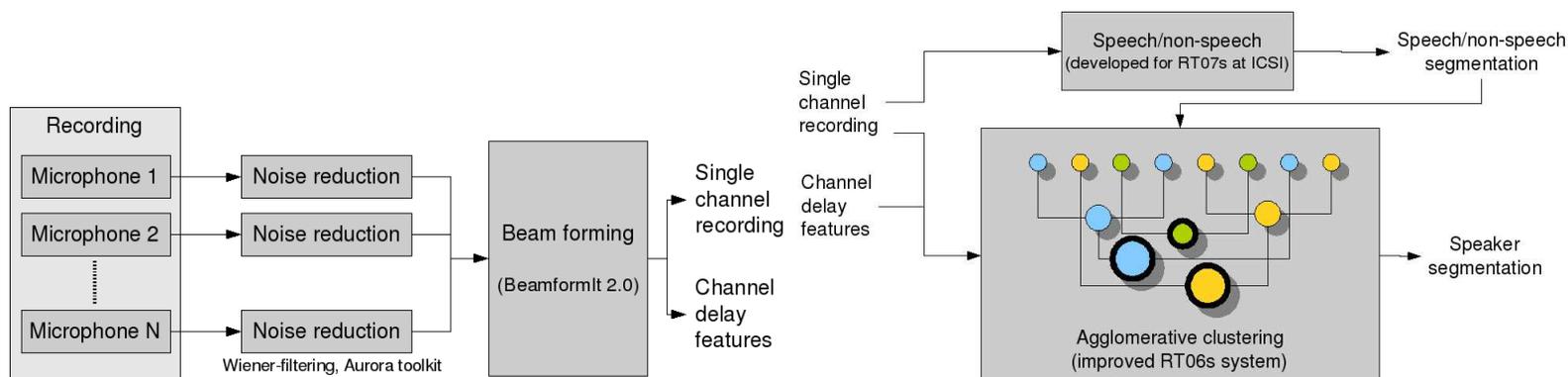
MDM front-end: Beamforming experiments

- ▶ Switched to using ICSI beamformer this year due to better stability

train beamforming	test beamforming	WER
AMI	AMI	42.6
AMI	ICSI-beam3.3	42.3
AMI	ICSI-beam2.0	41.9
ICSI-beam3.3	ICSI-beam3.3	41.5
ICSI-beam2.0	ICSI-beam2.0	40.4

- ▶ The cause for the discrepancy to the AMI beam-former is probably in the smoothing of delay estimates.

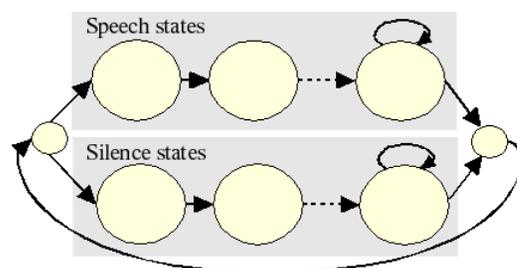
MDM front-end: segmentation and clustering



- ▶ The Aurora toolkit is used to apply Wiener-filtering on each individual channel.
- ▶ The BeamformIt toolkit (version 2.0) is used to combine the channels into one single channel.
- ▶ The delays between microphones, produced during beam-forming are stored for use by the speaker diarisation component.

MDM front-end: Speech activity detection

- ▶ The speech/non-speech component from the SHoUT toolkit is used.
- ▶ It is able to classify audio in three classes: silence, speech and audible non-speech
- ▶ It automatically generates models for these three classes using the audio it is processing
- ▶ Only rough baseline speech and silence models are needed for an initial bootstrap segmentation
- ▶ Speech and silence regions with high confidence are then used to train the models



MDM front-end: Speech activity detection results

- ▶ Experiments on the RT07s evaluation set.
 - ▷ speech/non-speech segmentation component compared to the reference segmentation.

%WER	P1	P3
Ref	42.1	36.3
SHoUT	43.8	38.1

sys07 results

SAD error rates	% missed speech	% false alarm	% SAD error
CMU_20061115-1030	2.60	4.40	7.05
CMU_20061115-1530	0.90	3.60	4.53
EDI_20061113-1500	1.00	1.20	2.16
EDI_20061114-1500	0.30	3.80	4.02
NIST_20051104-1515	0.10	0.40	0.57
NIST_20060216-1347	0.90	1.60	2.49
VT_20050408-1500	0.50	1.60	2.07
VT_20050425-1000	1.00	0.40	1.42
Overall error	0.90	2.10	3.06

Speaker diarisation

- ▶ Improved version of our RT06s AMI submission
 - ▷ Main improvement (except for bug-fixes): we have added a delay feature stream and
 - ▷ we have added a minimum duration constraint (250 states, 2.5 seconds)

Speaker segmentation	%WER RT07s P1	%WER RT07s pass 3
Reference speaker clustering	40.1	31.1
RT07s speaker clustering component	42.8	34.5
RT09s speaker clustering component	42.1	32.7

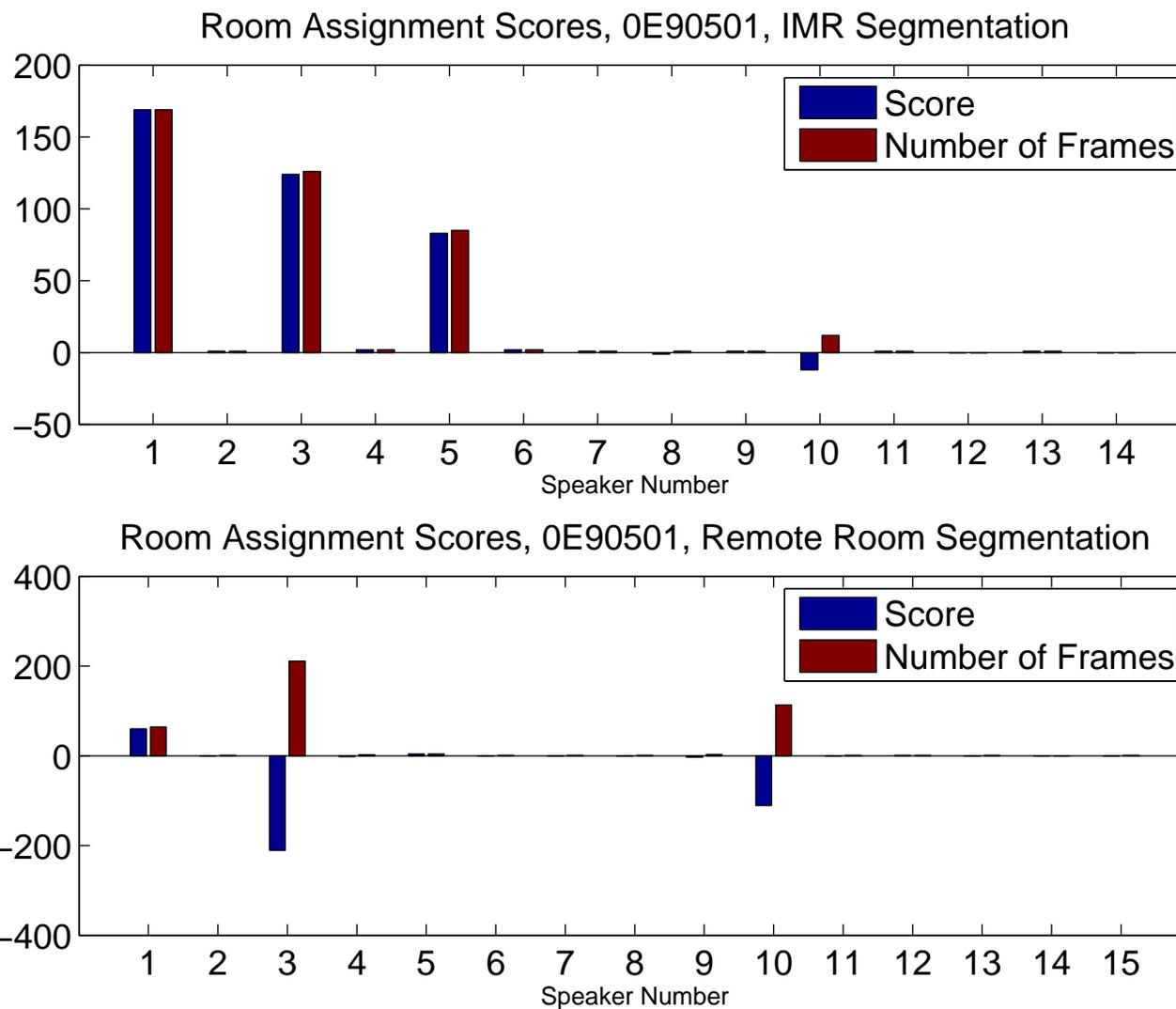
%WER on *rt07seval* MDM

MDM Front-end: Room Assignment

- ▶ Relies on delay in transmission of audio over video conference system
 1. Take beam-formed audio file for each room
 2. Perform speaker segmentation on room 1 audio
 3. For each speaker, for each frame, calculate the max of the cross correlation between the audio from room 1 and room 2 (i.e. the delay). +
If delay > 0 , increment room 1 count +
If delay < 0 , increment room 2 count
 4. Assign speaker to room with highest count
 5. Discard segments from speakers assigned to room 2
 6. Repeat using segmentation from room 2 audio, discarding segments assigned to room 1

- ▶ Large frame size (2.5 sec) used because of long delays in Video conference system

MDM front-end: Room Assignment - Counts



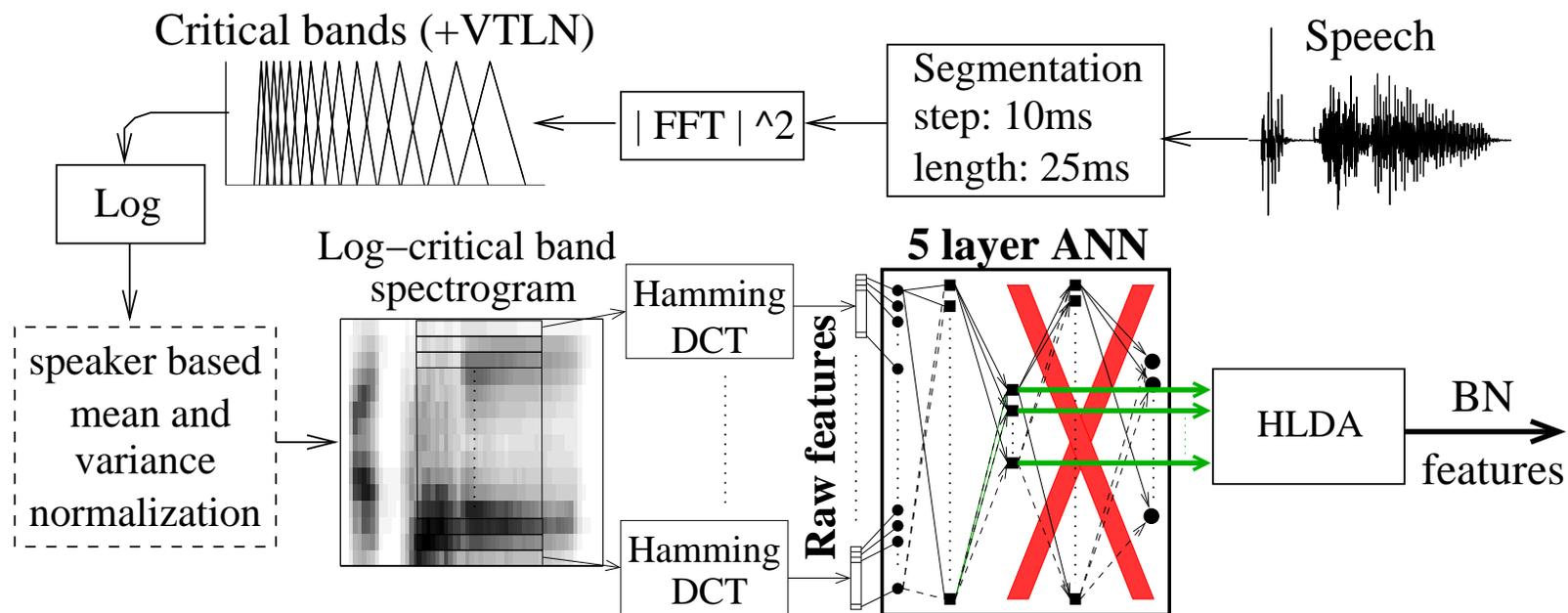
MDM front-end: Room assignment results

- ▶ Results only on *rt09eval*

Description	Segmentation	Tot	Sub	Del	Ins
room-assignment	Auto	33.2	20.6	9.3	3.2
only room1		36.3	20.5	12.7	3.1
only room2		45.1	25.8	14.8	4.4
ref. room-assignment	Ref	30.8	20.1	8.6	2.2
only room1		33.1	21.0	9.9	2.1
only room2		41.0	24.3	14.6	2.1

- ▶ Performance degradation due to segmentation is around 3% WER.
- ▶ Gain from room assignment is 3% if one considers “only room1” a valid baseline

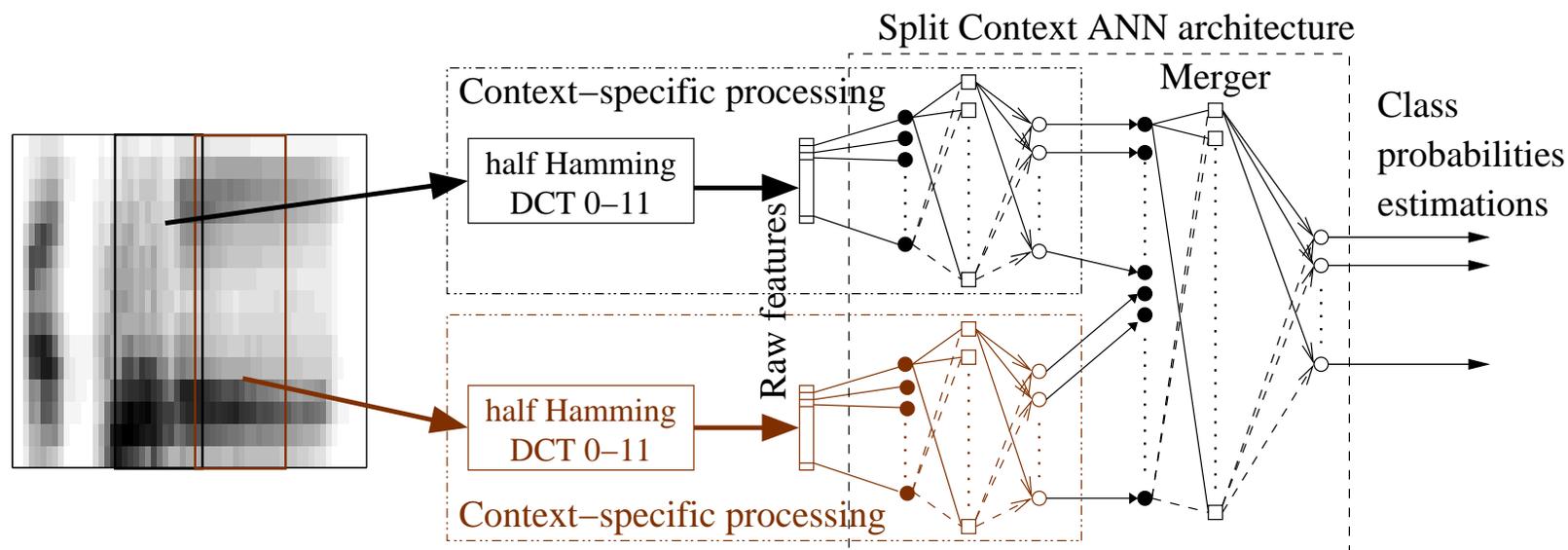
Front-end: Bottle-Neck Features



	WER
30h	25.2
180h	23.9

Amount of training data (Results on *rt05seval ref*)

IHM: NN based features - Split Context architecture

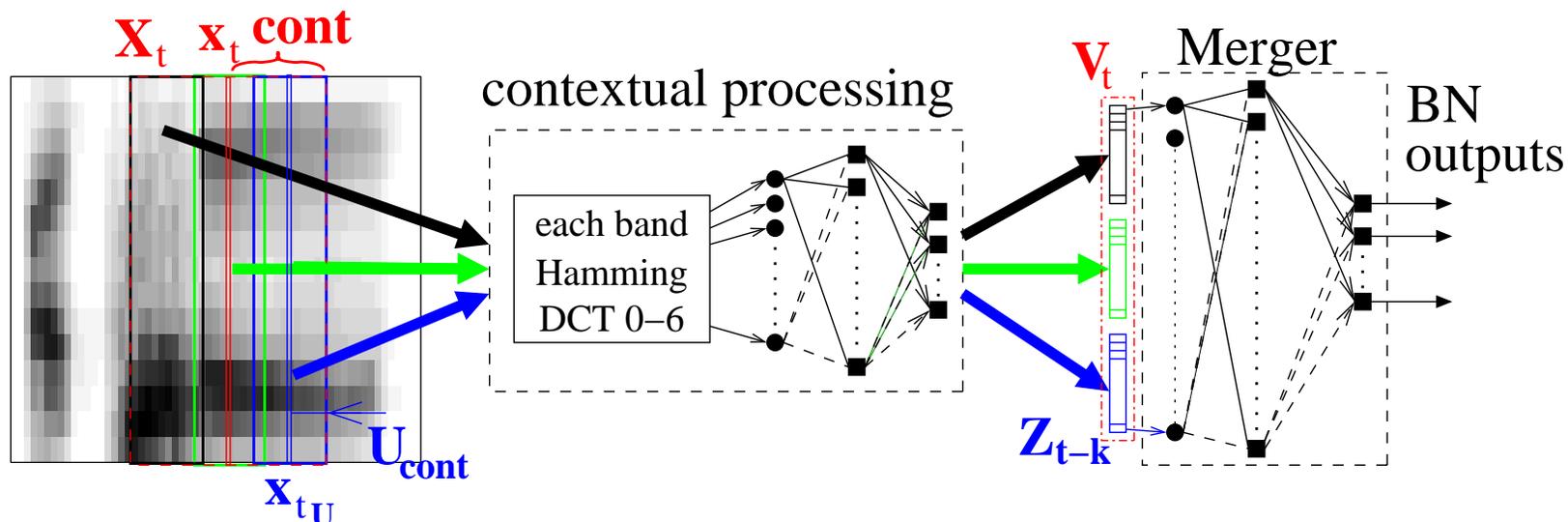


- Context NNs are standard probability estimators and merger had Bottle-neck structure (LCRC BN)

HLDA-PLP	36.0
HLDA-PLP + baseline BN	31.7
HLDA-PLP + LCRCBN	30.6

%WER on IHM *rt07sevalref*

IHM: NN based features - Stacked Bottleneck Architecture



- ▶ Processing of the smaller – contextual – block is done frame by frame and stacked, and only desired frames are taken to form merger input.
- ▷ The number of trainable parameters in the system is therefore reduced

Features	bottle-neck size				
	50	60	70	80	90
HLDA-PLP + SBN	29.5	29.4	29.5	29.4	29.4

fMPE and NN features

- ▶ RDLT implementation (Zhang 2006)
 - ▷ Posterior probabilities of the Gaussians are computed for each frame and these are spliced with the averages of posteriors for adjacent frames 1-2, 3-5 and 6-9 on the right and likewise for the left context (i.e. 7 groups spanning 19 frames in total).
 - ▷ All Gaussians in ML trained HMM model are pooled and clustered using agglomerative clustering to create GMM with 1000 components
 - ▷ only offset features (not the posteriors) are used

Features	Training			
	ML	MPE	fMPE	fMPE+MPE
HLDA-PLP	35.6	32.6	31.4	29.7
HLDA LCRCBN	30.4	28.1	26.7	26.3
HLDA-PLP +SBN	29.4	27.5	26.9	26.1
HLDA-PLP + LCRCBN + Δ	29.6	27.8	27.3	27.3

Complete Meeting Adaptation

▶ Idea

- ▷ Exploit the large amount of data per speaker by being selective

▶ Implementation

- ▷ Complimentary decoding with different LMs and acoustic models
- ▷ Align output with one acoustic model set
- ▷ Only keep words that occur in both transcripts with the same timing
- ▷ Use for adaptation

▶ Result

- ▷ **Discards** approximately **2.7 hours of data from 5.5**, but error rates on those parts is low
- ▷ Full meeting adaptation brings 0.2% WER gain, but intersection of outputs does not improve on that (yet).

Adaptation experiments

CMLLR	MLLR	Tot	Sub	Del	Ins	IDI	EDI	NIST
-	-	26.3	17.7	5.3	3.3	30.3	23.1	25.3
2	-	24.6	16.0	5.9	2.6	28.2	21.4	23.9
4	-	24.2	15.8	5.9	2.5	27.8	21.1	23.5
6	-	24.2	15.8	5.9	2.5	27.9	20.8	23.5
8	-	24.6	16.0	6.1	2.5	28.1	21.0	24.3
4	32D	24.1	15.8	5.8	2.5	27.6	21.0	23.4
4	16D	24.1	15.8	5.7	2.5	27.6	21.1	23.3
4	8D	24.1	15.8	5.8	2.5	27.5	21.1	23.3
4	4D	24.2	15.8	5.8	2.5	27.6	21.2	23.4
4	32F	27.2	17.8	6.5	2.8	30.5	22.7	27.7
4	16F	25.6	16.7	6.3	2.6	28.8	21.0	26.3
4	8F	24.4	15.9	6.0	2.5	28.0	20.5	24.4
4	4F	24.1	15.7	5.9	2.5	27.6	20.7	23.6

Results on *rt09eval* IHM, M3 models

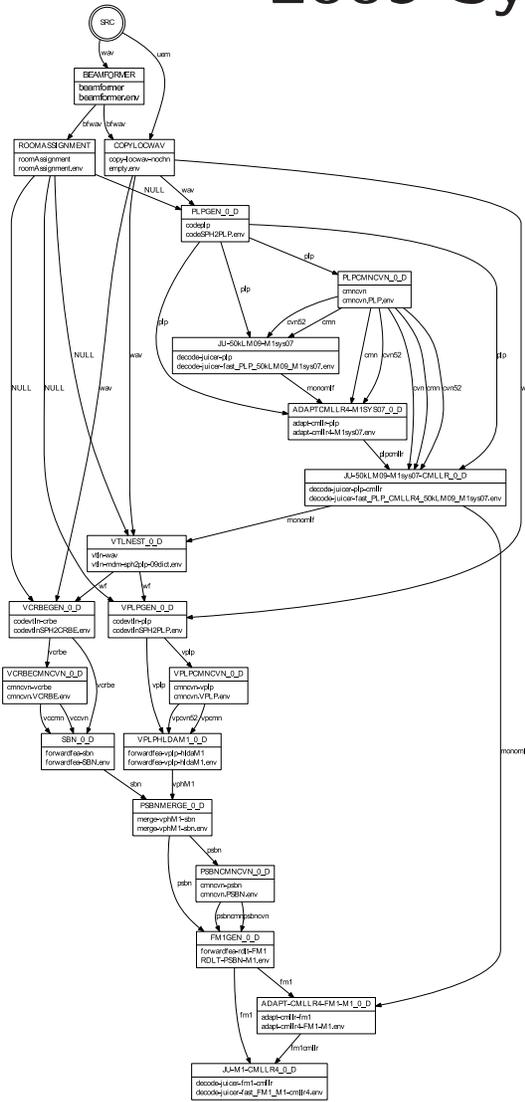
2009 IHM System - Stages

1. Segmentation
2. Initial decoding of full meeting with
 - (a) 4g LM based on 50K vocabulary and weak acoustic model (ML) **M1**
 - (b) 7g LM based on 6K vocabulary and strong acoustic model (MPE) **M2**
3. Intersect output and adapt (CMLLR)
4. Decode using M2 models and 4gLM on 50k vocabulary
5. Compute VTLN/SBN/fMPE
6. Adapt SBN/fMPE/MPE models **M3** using CMLLR
7. Adapt LCRCBN/fMPE/MPE models **M4** using CMLLR and output of previous stage
8. Generate 4g lattices with adapted M4 models
9. Rescore using M1 models and CMLLR + MLLR adaptation
10. Compute Confusion networks

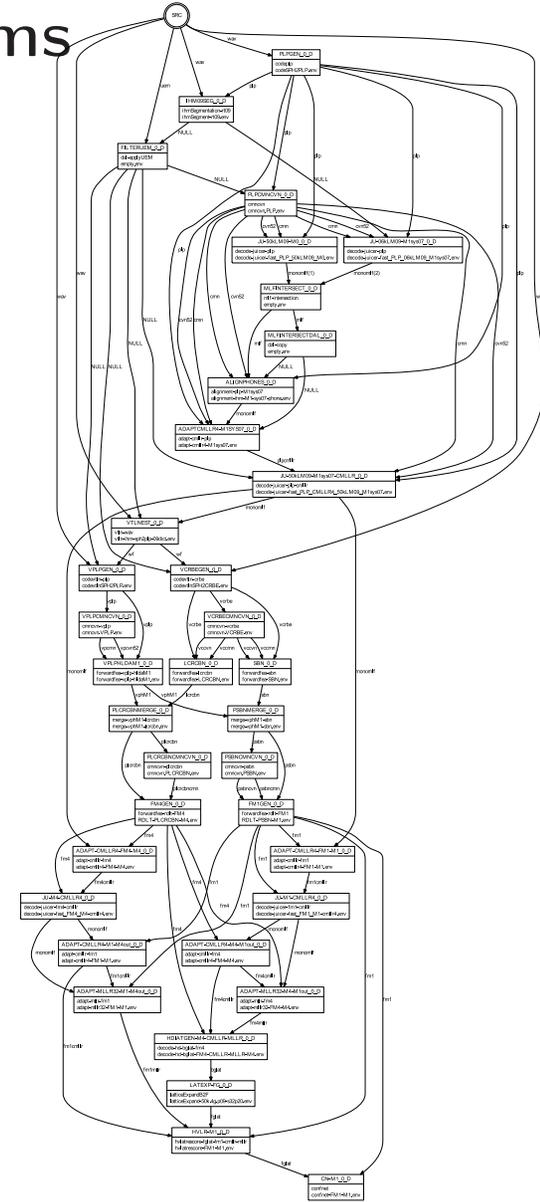
2009 MDM System - Stages

1. Beamforming
2. Segmentation and Clustering
3. Room Assignment
4. Initial decode using 4g LM and 50k vocabulary and MPE models (2007 P1 models)
5. Decode using M2 models and 4gLM on 50k vocabulary
6. Compute VTLN/SBN/fMPE
7. Adapt SBN/fMPE/MPE models using CMLLR
8. Decode using the SBN/fMPE/MPE adapted models
9. (SASTT) Apply AMI diarisation

2009 Systems - Diagrams



MDM



IHM

Results *rt07seval* IHM - Automatic Segmentation

Description	Tot	Sub	Del	Ins	CMU	EDI	NIST	VT
6kLM M2	38.1	22.5	9.8	5.8	51.7	30.2	32.5	37.5
50kLM09 M1	40.7	24.4	9.0	7.3	55.6	33.6	34.6	38.1
50kLM09 M2 CMLLR	32.1	18.4	9.5	4.3	43.2	25.2	27.3	32.0
M3 CMLLR	24.5	13.1	8.4	3.0	35.1	18.1	20.5	23.9
Latgen BG	26.7	14.4	9.2	3.0	37.1	20.6	22.0	26.6
Lat exp 4g	23.8	12.6	8.3	3.0	34.3	17.8	19.5	23.4
Lat rescore M3	23.5	12.5	8.0	3.0	33.9	17.2	19.5	22.8
cn	23.4	12.7	7.5	3.2	33.8	17.5	19.1	22.8

- ▶ Tried quite a view other strategies for system combination and rescoreing but none gave any improvement

Results *rt09eval* IHM - Automatic Segmentation

Description	Tot	Sub	Del	Ins	IDI	EDI	NIST
6kLM M2	41.3	24.0	12.0	5.3	45.1	32.3	44.9
50kLM09 M1	45.9	27.5	11.4	6.9	50.9	36.8	48.3
50kLM09 M2 CMLLR	36.4	20.6	11.9	3.9	38.8	28.5	40.2
M3 CMLLR	28.3	14.6	11.0	2.7	28.5	21.4	33.2
Lat Gen 2g	30.3	16.0	11.4	2.9	31.6	23.0	34.7
Lat Exp 4g	27.6	14.1	10.7	2.7	28.3	20.9	31.9
Lat Res M3	27.2	14.0	10.6	2.7	28.0	20.3	31.9
cn	27.4	14.3	10.0	3.1	28.6	20.4	31.6

Results *rt09eval* - Reference Segmentation

- Several minor bug-fixes to the official system

Description	Tot	Sub	Del	Ins	IDI	EDI	NIST
6kLM M2	38.3	25.8	7.3	5.2	44.0	31.9	38.3
50kLM09 M1	43.7	30.1	6.4	7.3	50.2	36.8	43.3
50kLM09 M2 CMLLR	32.9	22.1	7.1	3.7	37.9	27.7	32.5
M3 CMLLR	24.2	15.8	5.9	2.5	27.8	21.1	23.5
Lat Gen 2g	26.7	17.4	6.6	2.7	31.0	23.1	25.7
Lat Exp 4g	23.9	15.4	5.9	2.5	27.9	20.6	22.8
Lat Res M3	23.5	15.3	5.6	2.5	27.5	20.0	22.6
cn	23.8	15.8	5.0	3.0	28.0	20.7	22.5

- Exceptional gains froms adaptation on IDI data

Results MDM - Automatic Segmentation

	rt07seval				rt09eval			
Automatic segmentation	Tot	Sub	Del	Ins	Tot	Sub	Del	Ins
Initial	40.3	25.1	11.1	4.2	44.2	28.7	10.8	4.7
Final	29.3	17.0	9.0	3.3	33.2	20.6	9.3	3.2
Reference segmentation	rt07seval				rt09eval			
Initial	37.8	24.4	11.1	2.3	42.3	28.8	10.3	3.2
Final	26.5	16.4	8.4	1.6	30.7	20.3	8.3	2.1

- ▶ 3% lost due to segmentation/clustering
- ▶ *rt09eval* is much harder than

RTF on *rt09eval*

	RTF	RTF with loading	WER
Full meeting adaptation	4.57	5.16	-
Adapted first pass (PLP)	5.86	6.72	32.9
Adapted second pass (FM1)	8.66	9.84	24.2
Cross-adaptation HDecode (lat. gen.)	17.95	19.44	23.8
Cross-adaptation Juicer	12.83	14.57	23.8

► Throughput using ROTK

- ▷ Automatic distribution of computing jobs accounting to algorithmic dependencies
- ▷ On an empty 80 node cluster allows processing of the *rt09eval* set in approx. 4 hours (slightly above 1 RTF)
- ▷ Naturally the cluster utilisation is around 20%

Conclusions/Summary

- ▶ Improvements targeting both performance and speed
 - ▷ fMPE gives significant improvement
 - ▷ New NN architectures give modestly better results.
 - ▷ IHM segmentation requires better robustness !
 - ▷ MDM segmentation worked very well
 - ▷ Juicer is now competitive in speed and performance
 - ▷ New language models
 - ▷ All systems operated with 20 RTF, but 10RTF performance is very close.

- ▶ Still no specific handling of overlap

- ▶ The AMI system is online at www.webasr.org