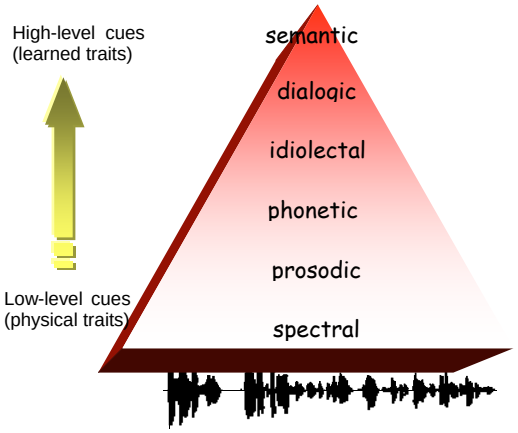
MIT Lincoln Laboratory System Description
NIST SRE 2006

Bill Campbell, Doug Sturim, Wade Shen, Doug Reynolds
MIT Lincoln Laboratory
Jiri Navratil
IBM

1 INTRODUCTION

The JHU SuperSID workshop in 2002 (http://www.clsp.jhu.edu/ws2002/groups/supersid/) demonstrated the power of exploiting multiple levels of information conveying speaker identity in the speech signal. As illustrated in the iconic speaker-feature pyramid, features and classifiers worked from low-level spectra to high-level words. While the high-level features from prosodics, phones and words have indeed provided accuracy improvements in past speaker recognition evaluations, it has been well established that spectral based features and classifiers provide the main foundation of system accuracy where the vexing problems of channel variability can be directly addressed. Further, the incremental gains in accuracy obtained from high-level cues often come at significant increases in computational and processing complexity. With these issues in mind, MITLL's focus for SRE-2006 was on ***building the base*** by emphasizing spectral based systems which offer computational speed, show robustness to channel variability, are immune to language and thus are well suited for porting to new environments and novel applications.



The MIT Lincoln Laboratory submissions for the 2006 NIST Speaker Recognition Evaluation (SRE) are built using the systems listed in Table 1 and described in the following sections. The table also lists a rough order of processing time for each system. Systems are combined using an MLP based fusion system.

Table 1: Core systems used in the MITLL SRE06 submissions

System	Features	Classifier	Scoring Factor (faster than speech time[1])
GMM-ATNORM	MFCC	GMM-UBM, w/ ATNORM cohort selection	9.6 (1 model) 7.9 (55 T-norm)
GMM-LFA	MFCC	GMM-UBM w/ Latent Factor Analysis	2.7 (1 model) 2.4 (100 T-norm)
SVM-GLDS	MFCC & LPCC	SVM GLDS kernel w/ Nuisance Attribute Projection	274 (1 model)
SVM-GSV	SuperVector of GMM means	SVM linear kernel w/ Nuisance Attribute Projection	173 (1 model)
SVM-MLLR	MLLR parameters from BYBLOS STT	SVM linear kernel w/ Nuisance Attribute Projection	0.83 (inc. first-pass STT)
SVM-WORD	Word lattice from BYBLOS STT	SVM weighted linear kernel	0.45 (inc. STT)
BT-WORD (IBM)	Words from BYBLOS STT	IBM's Binary Tree	0.45 (inc. STT)

[1] Scoring Factor = (speech duration)/(user+sys time) computed for R&D system configuration on 2.8-3.0 GHz Linux processor. System timing includes feature extraction from audio file. STT processing factor is approx. 0.45.

NGRAM-WORD	Word lattice from BYBLOS STT	Ngram LMs	0.18 (inc. STT and T-norm)
SVM-WORD_DUR	Word durations from BYBLOS STT	SVM with	0.43 (inc. STT and T-norm)

2 Training and Development Data Utilized

Data to train explicit core system parameters (background models, T-norm models, Z-norm data, LFA subspace projections, NAP projections) were obtained from: Switchboard II phases 1-5, SRE04 (Mixer), and Fisher. The only exception is the NAP projection used for the 1convmic condition was derived using the SRE05 cross-channel data. Development testing and fusion parameters were obtained using the SRE05 test sets.

3 Spectral Based Systems

3.1 Speech Activity Detection

The spectral-based systems used a common set of speech activity detection marks from a GMM based SAD system. The GMM-SAD system uses two 128 mixture GMMs: one trained using speech from Switchboard files and one trained from non-speech segments of Switchboard, hold music and telephone sounds (rings, tone, etc.). Per-frame likelihood ratio scores are smoothed using a 0.5 second window and speech segments are detected with a threshold of 0, with no additional filtering. For some systems an additional adaptive energy-based SAD is run on the output of the GMM-SAD marks.

3.2 GMM-ATNORM

The MITLL GMM-UBM speaker detection system, fully described in [1], is similar to that used in previous evaluations. The main differences this year are

- A GMM based speech detector was used as initial speech detector followed by a second stage energy based speech detector.
- The UBM was trained using Switchboard II and SRE04 corpora

Techniques to deal with cross-language conditions, such as feature domain language mapping and speaker-dependent language specific UBMs, were experimented with but showed no improvements on the dev data.

A 19-dimensional mel-cepstral vector is extracted from the speech signal every 10ms using a 20ms window. The mel-cepstral vector is computed using a simulated triangular filterbank on the DFT spectrum. Bandlimitng is then performed by only retaining the filterbank outputs from the frequency range 300Hz-3138Hz. Cepstral vectors are processed with RASTA filtering to mitigate linear channel bias effects. Delta cepstral are then computed over a +-2 frame span and appended to the cepstra vector producing a 38 dimensional feature vector. The feature vector stream is then processed through an adaptive, energy-based speech detector to discard low-energy vectors. The silence removed features are processed with feature mapping [2] and, finally, normalized by removing the global mean and dividing by the standard deviation. All processing steps are performed on the fly during processing (i.e., processing occurred directly from sph file data) and are included in reported processing times.

T-norm [3] is a technique where scores from a collection of fixed non-target models are used to normalize a target model score for a test file. The target model score normalization is accomplished by subtracting the mean and dividing by the standard deviation of the non-target model scores per test file. The T-norm set, derived from the SRE04 corpora, used this year consisted of:

- 8conv4w: 224 female and 170 male speakers
- 3conv4w: 269 female and 179 male speakers
- 1conv4w: 364 female and 243 male speakers

Speaker-specific T-norm selection (also known as cohort selection) was used [4]. The speaker-dependent T-norm set was determined by the T-norm models which scored the most similar as the speaker model on a set of imposter utterance. The cohort models are selected by testing all cohort models and the target model with the same set of N impostor-test segments. The output scores are formed into M+1 vectors of dimension N (cohorts plus the target model). Euclidean distances are calculated from the cohorts to the target model. The closest set of P cohort models are used to T-norm during run time. For this system P was empirically chosen to be 55.

3.3 GMM-LFA

The GMM Latent factor analysis system (LFA) was based directly on the work presented in [5]. The approach models session variability through a low dimensional subspace projection in both training and testing. The session variability is modeled as an low-dimensional additive bias to the model means:

$$m_i(s) = m(s) + U\, x(s).$$

where $m_i(s)$ and $m(s)$ are "supervectors" of stacked means GMM means [6] and [5]. The $m_i(s)$ is the supervector from the i-th session of talker "s" whereas the m(s) is the session independent term of talker s. The GMM-UBM supervectors were generated with the same system used in the Atnorm system described earlier.

Training of the low-rank transformation matrix U was generated directly as described in [7] and not iteratively. The datasets used to train the low-rank transformation matrix were the Switchboard II phase 1-5 corpus.

Z-norm followed by T-normalization was also performed on the scores. The Z-norm imposter test messages were drawn from the Switchboard II phase 1-5 corpus. The breakdown of cohorts, drawn from the SRE04 and Fisher corpora, were as follows:

- 8conv4w: 224 female and 170 male speakers
- 3conv4w: 269 female and 179 male speakers
- 1conv4w: 664 female and 543 male speakers

3.4 SVM-GLDS

The SVM GLDS system uses the polynomial based kernel described in [8]. We used a degree 3 basis of monomials.

The SVM GLDS uses two front ends. One front end is based on 19 MFCCs plus deltas. RASTA, mean, and variance normalization are applied to the features. A second front end is based on 18 LPCCs plus deltas; the LPCCs are obtained from 12 LP coefficients.

Backgrounds for the SVM GLDS systems were obtained from a subset of the English portion of the Fisher corpus. The NAP projection [9] for telephone data was trained on the Switchboard 2 corpus. For conversational microphone data, the NAP projection was trained on SRE05 microphone evaluation data and corresponding telephone data.

NAP projection was applied in training to the backgrounds and speaker data. SVM models were obtained using the GLDS kernel and SVM Light. Model compaction was used to reduce the size of models.

For scoring, we computed an inner product between the average expansion and the SVM model. The scores from the two feature sets were combined in equal weights using linear fusion. No additional normalization was applied.

3.5 SVM-GSV

The SVM GMM supervector (GSV) system uses a novel kernel based upon an approximation to the KL divergence. This method is described in detail in the [10,11].

Feature extraction is performed using a standard MFCC front-end with 19 features plus deltas. RASTA, CMS, and variance normalization are applied to the features. A 2048 mixture GMM UBM is used to obtain the GMM supervectors. A relevance factor of 16 is used in the MAP adaptation.

The SVM GSV background was computed from a subset of the Fisher English corpora. A session NAP projection was trained using the same set as the SVM GLDS kernel. The projection was applied to both speaker training data and background data. Training was performed using pre-computed kernel inner products with the SVMTorch tool. Models were compacted to reduce compute time and storage.

Scoring was performed by doing a one-pass MAP adaptation on the test utterance and then scoring using an inner product. No additional normalization was performed.

4 STT Based Systems

4.1 Byblos STT

We extracted word lattices and MLLR parameters using BBN's Byblos 1xRT recognizer trained with 2000+ hours of telephone speech [12]. Audio files were first segmented into chunks of 15 seconds or less using a two-class HMM (speech/non-speech) trained on a small selection (approx. four hours) of Switchboard II and Fisher data. Word lattices are generated for each segment using Byblos (with SCTM + VTLN and HLDA adaptation). MLLR parameters are obtained after the un-adapted decode pass of the recognizer.

4.2 SVM-MLLR

We used the MLLR transforms from the BBN Byblos STT recognizer as features for an SVM-based speaker recognition system. The approach we took is based on the work described in [13] with a number of minor differences:

1. We use two gender-independent regression classes and a global transform for features. The final dimensionality of each feature vector was 10980.
2. We apply 0-1 normalization to the each feature vector using statistics derived from the background model.
3. We apply Nuisance Attribute Projection as described in [9].

During testing T-norm is optionally applied.

4.3 SVM-WORD

The SVM word system use a kernel for comparing conversation sides based upon methods from information retrieval. Sequences of tokens are converted to a vector of probabilities of occurrences of terms and co-occurrences of terms (bag of unigram and bag of bigrams). This method was first used in the NIST SRE 2003 evaluation and is documented in [14]. Weighting for the word system was based upon a log() penalty of the inverse background probability of an *n*-gram.

Speech to text output was obtained from the BBN Byblos system. Expected counts and probabilities of *n*-grams were calculated using SRI's language modeling tool [15]. These probabilities were then stored in a sparse vector.

The SVM used a weighted linear kernel [14]. This amounted to scaling individual entries in the vector of probabilities with a term weighting of $f(1/p(t_i))$, where $p(t_i)$ was the probability of the term over all conversations in the background.

The background set used for training the SVM was derived from Fisher English, Arabic, and Mandarin data. SVM training was performed using SVM Light.

For the SVM word system, probabilities were derived using expected counts and then weighted appropriately. Scoring was performed using a linear kernel with the target speaker model.

4.4 BT-WORD (IBM)

The BT model consists of a set of non-terminal and a set of terminal nodes. Each non-terminal node is associated with a binary test and has two child nodes; each terminal node (leaf) contains a token distribution and has no child nodes. In order to calculate the probability of a token at time t, given a certain history of tokens a_{t-1},...,a_{t-k} (referred to as predictors), the tree structure is traversed top-down via non-terminal nodes with the path being determined by outcomes of binary tests (questions) until a terminal node is reached and the probability of the token a_t can be determined from the leaf distribution. An example of a binary question may be "Is predictor a_{t-3} in set {[a],[oe],[e]} ? ". Since the path through the tree is determined by the predictors, i.e. token context, the token history is modeled in a flexible way allowing for a varying degree of complexity in clustering the space of all token histories. The crux of the BT modeling task is building an appropriate speaker tree, namely determining the node questions as well as leaf distributions. In this system a minimum-prediction entropy criterion (corresponding to an ML criterion) was used

In this evaluation, the STT transcripts of the speaker speech were used to generate sequences of tokens with an inventory defined as the 512 top frequent words plus an additional "other" class representing the remaining STT vocabulary.

The tree structures in this evaluation were trained using a fast flip-flop algorithm to minimize the prediction entropy in terminal nodes as described in [16].

First, on data from background speakers (**BKG** data set), a common BT model was created resulting in a BT with about 15k terminal nodes using up to 2 predictors (i.e. exploiting a context of 3 words at a time). Subsequently, individual target speaker models were created using an adaptive BT training algorithm from the common BT model as described in [17].

The probability of a token a_t in a sequence generated by the ASR tokenizer and given a speaker hypothesis S_j, is retrieved from the corresponding BT model in a way described above (traversing the tree). In addition, a recursive parental-node smoothing is applied to the probability as described in [17]. The resulting BT score is S(a) = \sum_t p_{BT}(a_t| Pred(a_t))/T , where a=a_1,...,a_T is the token sequence.

A C-norm [18] (a variant of the H-norm) followed by the T-Norm standardization is applied to the scores. The C-norm is based on an automatic gender-dependent 5-channel detector (identical to [19]). The score normalization is applied as follows:

- S_c(a) = (S(a) - m_{cj})/h_{cj} , with m_{cj} and h_{cj} denoting the mean and standard deviation of scores from channel c given speaker model j
- S_{ct}(a)=(S_c(a)-m_t)/h_t, with m_t, h_t denoting the mean and std. deviation of scores of the test on the T-norm speakers (after C-norm)

4.5 NGRAM-WORD

The word n-gram system attempts to recognize the speaker using n-gram frequency information. Word are extracted from an utterance by using the BBN Byblos 1xRT STT system to generate lattices. From these lattices expected counts of n-grams are computed by estimating n-gram posteriors (using a standard forward-backward approach in SRI's lattice-tool).

During scoring, these expected counts are used to compute p_model(w_i, w_{i-1}), p_message(w_i, w_{i-1}) and p_bkg(w_i, w_{i-1}). These probabilities are then used to compute the cross perplexity between the test message and the target and background models. The final score is the ratio of the target model score and the background score. ZT-norm was applied to these scores using models and non-target messages trained from SRE-04.S

4.6 SVM-WORD_DUR

The word duration system we implemented for speaker recognition models the expected duration of phones in words. Each utterance is represented as a feature vector of ~56,000 phone durations in word contexts (those seen in our background training set, a ~3,000 utterance subset of the Fisher Corpus) is constructed.

During training, vectors from a target speaker are used in conjunction with vectors for a background model to train an SVM. We apply a relative expected duration kernel, normalizing by the expected phone duration of each phone (in word context) from our background model.

Each message model pair is scored as a weighted inner product between model support vectors and the test message. T-norm is then optionally applied to the resulting scores using models trained from SRE-04.

5 Multi-Speaker Speech Processing

For multi-speaker speech processing we applied automatic segmentation and clustering to purify the train and test data which was then processed by the core detection systems.

Speaker segmentation and clustering was performed using the following steps. First, a speech activity detector parses the speech into speech and silence segments. Within the speech segments, a speaker change detector finds putative change points. The segments from the change detector are then sent into an agglomerative clustering system using a full-covariance Gaussian model per-cluster and a BIC-based stopping criterion. The output from the cluster is used to train GMMs and the file is iteratively re-segmented. More details of the diarization system can be found in [20,21,22].

In 3conv2w training, the output clusters from the three training conversations are further processed by an agglomerative clustering system. This final clustering uses a cross-likelihood ratio distance between all clusters and the condition that only one cluster can come from each training speech file. The final set with the smallest inter-cluster distance is selected for training.

In 1conv2w testing, the conversation was automatically clustered, the individual clusters were processed via the 1sp systems and the maximum score for each message cluster was selected.

6 System Fusion

The scores from the systems were fused with a perceptron classifier using LNKnet [23]. The perceptron architecture chosen has N input nodes, no hidden layers, and two output nodes. Input values to the perceptron were normalized to zero mean and unit standard deviation using parameters derived from the training data. The perceptron weights were trained using the entire development data with a mean squared error criteria. The classifier corresponding to the number of training conversations is then used to fuse scores from systems. The fusion classifier is trained to minimum the DCF by using prior probability for the target class in training and testing set to 0.09 corresponding to the costs and priors (C_miss*P_tgt/(Cmiss*P_tgt + C_fa*(1-P_tgt))). The score for the test file was then remapped to the application prior of 0.01. The minDCF threshold from cross-validation experiments on the development data were used to make hard decision for the submissions.

Development experiment using external metadata as inputs to the fuser as proposed in [24] found little gain so were not used for the submission systems.

7 Submission Systems

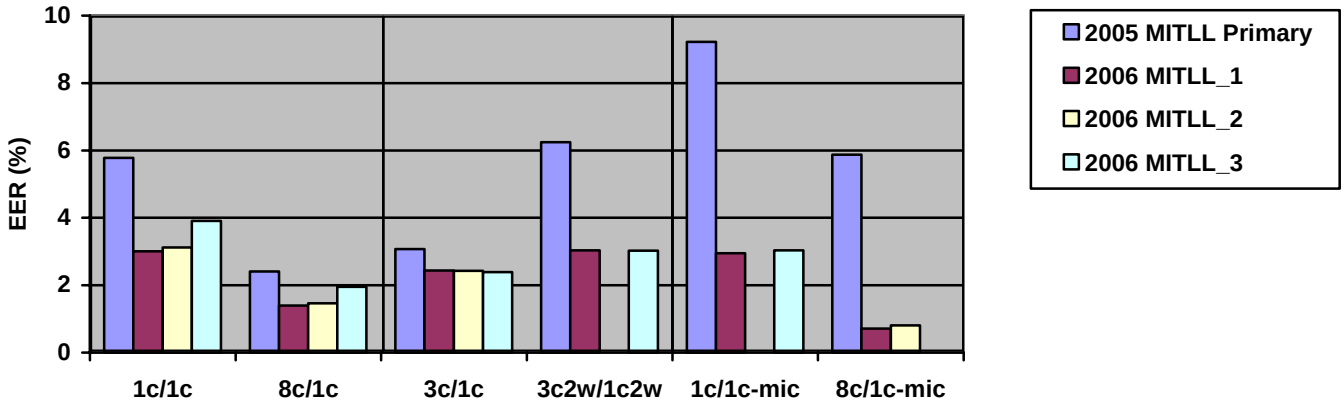
The core systems used in the submission combinations for the MITLL submissions are given in the following table. The fuser was trained using only the SRE05 common condition trials for MITLL_1 and MITLL_3. All SRE05 trails were used for training the fuser in MITLL_2. The systems included in MITLL_1 were selected as the set giving the minimum DCF based on development experiments. Generally MITLL_2 is a contrast using all systems available for a condition. And MITLL_3 is a combination of only spectral-based systems.

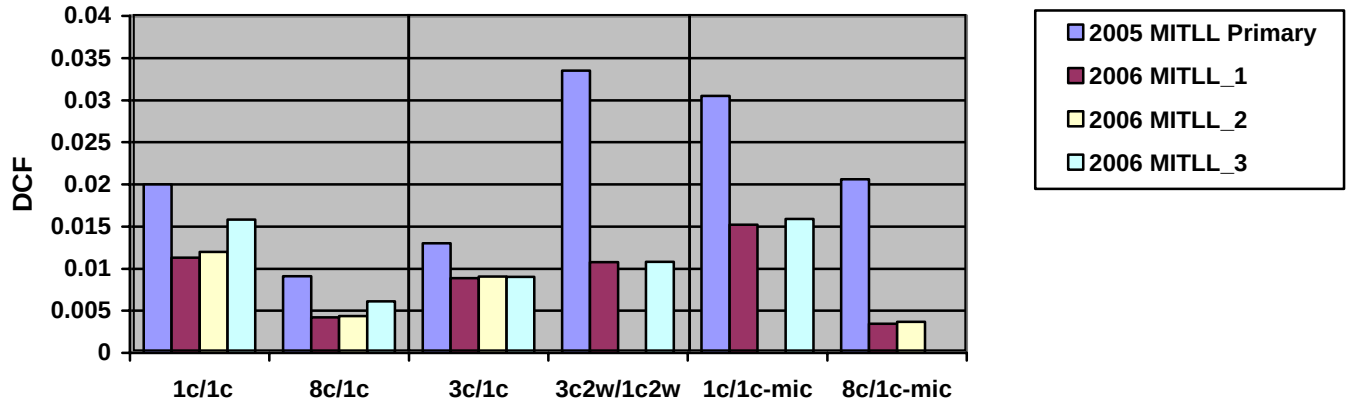
Table 2: Systems comprising the MITLL SRE06 submissions

Condition	Submission	GMM-ATNORM	GMM-LFA	SVM-GLDS	SVM-GSV	SVM-MLLR	SVM-WORD	BT-WORD	NGRAM-WORD	SVM-WORD_DUR
1c/1c	MITLL_1	X	X	X	X	X	X	X		
	MITLL_2	X	X	X	X	X	X	X		
	MITLL_3	X	X	X	X					
3c/1c	MITLL_1		X	X	X					
	MITLL_2	X	X	X	X					
	MITLL_3	X	X	X	X					
8c/1c	MITLL_1		X	X	X	X	X	X	X	
	MITLL_2	X	X	X	X	X	X	X	X	X
	MITLL_3	X	X	X	X					
3c2w/ 1c2w	MITLL_1		X	X	X					
	MITLL_2									
	MITLL_3	X	X	X	X					
1c/1c-mic	MITLL_1			X	X					
	MITLL_2									
	MITLL_3		X	X	X					
8c/8c-mic	MITLL_1		X	X	X					
	MITLL_2			X						
	MITLL_3									

8 Development Data Results

In the following charts we show EER and minimum DCF on the SRE05 data using the MITLL 2005 primary system and the 2006 submission systems. Results are from the common trials for all conditions except the microphone tests. Overall the 2006 systems showed significant gains, with generally 50% or greater reductions in EER and/or DCF. Again we have found that the spectral based systems are the main performance driver and can produce very low error rates with low processing and complexity.





Acknowledgments

We are glad to have the opportunity to partner with IBM Research via Jiri Navratil of the Conversational Biometrics Group. We also wish to thank BBN for allowing the use of their Byblos STT system.

References

[1] D. A. Reynolds, T. F. Quatieri and R. B. Dunn, Speaker Verification Using Adapted Gaussian Mixture Models, Digital Signal Processing, 10 (2000), pp. 19-41.

[2] D. A. Reynolds, Channel Robust Speaker Verification via Feature Mapping, International Conference on Acoustics, Speech, and Signal Processing, IEEE, Hong Kong, 2003, pp. 53-56.

[3] R. Auckenthaler, M. Carey and H. Lloyd-Thomas, Score Normalization for Text-Independent Speaker Verification Systems, Digital Signal Processing, 10 (2000), pp. 42-54.

[4] D. E. Sturim and D. A. Reynolds, "Speaker Adaptive Cohort Selection for Tnorm in Text-Independent Speaker Verification," ICASSP 2005.

[5] R. Vogt, B. Baker, S. Sridharan, Modelling Session Variability in Text-Independent Speaker Verification, EuroSpeech 2006.

[6] P. Kenny, G. Boulianne, and P. Dumouchel, "Eigenvoice Modeling With Sparse Training Data", IEEE Transactions On Speech And Audio Processing, Vol. 13, No. 3, May 2005 345.

[7] M. Tipping and C. Bishop, Mixtures of probabilistic principal component analyzers, Neural Computation, vol. 11, pp. 435, 1999

[8] W. M. Campbell, "A SVM/HMM System for Speaker Recognition," Proc. International Conference on Acoustics, Speech, and Signal Processing, Hong Kong, 2003, pp. 209-302.

[9] A. Solomonoff, W. M. Campbell, I. Boardman, "Advances In Channel Compensation for SVM Speaker Recognition," Proc. International Conference on Acoustics, Speech, and Signal Processing, Philadelphia, PA, 2005.

[10] W. M. Campbell, D. E. Sturim, and D. A. Reynolds, "Support Vector Machines Using GMM Supervectors for Speaker Verification," IEEE Signal Processing Letters, May 2006, Vol 13, no. 5, pp 308-311.

[11] W. M. Campbell, D. E. Sturim, D. A. Reynolds, and A. Solomonoff, "SVM Based Speaker Verification Using a GMM Supervector and NAP Variability Compensation," Proc. ICASSP 2006.

[12] S. Matsoukas, R. Prasad, B. Xiang, L. Nguyen, and R. Schwartz, "The RT04 BBN 1xRT Recognition Systems for English CTS and BN," in Proceedings of Rich Transcription Workshop, Palisades, NY, Nov. 2004 (http://www.sainc.com/richtrans2004/uploads/wednesday/The%20RT04%20BBN%201xRT%20Recognition%20Systems.pdf)

[13] A. Stolcke, L. Ferrer, S. Kajarekar, E. Shriberg, & A. Venkataraman, MLLR Transforms as Features in Speaker Recognition. *Proc. Eurospeech*, Lisbon, pp. 2425-2428

[14] W. M. Campbell, D. A. Reynolds, D. A. Jones, and T. R. Leek., "High-Level Speaker Verification with Support Vector Machines," in Proc. International Conference on Acoustics, Speech, and Signal Processing in Montréal, Québec, Canada, pp. I: 73-76, 17-21 May 2004.

[15] A. Hatch, B. Peskin, and A. Stolcke, "Improved Phonetic Speaker Recognition Using Lattice Decoding," International Conference on Acoustics, Speech and Signal Processing, Philadelphia, PA, 2005.

[16] J. Navratil ``Recent advances in phonotactic language recognition using binary decision trees ," submitted to Interspeech 2006.
[17] J. Navratil, Q. Jin, W. Andrews, J.P. Campbell, ``Phonetic speaker recognition using maximum-likelihood binary-decision tree models," ICASSP-2003, Hong Kong, April, 2003.
[18] D. Reynolds, "Comparison of Background Normalization Methods for Text-Independent Speaker Verification", in proc. of Eurospeech, pp 963-966, vol 2, 1997.
[19] J. Pelecanos et al., "The IBM NIST SRE05 System Description," NIST SRE05 Workshop, June, 2005
[20] R. B. Dunn, D. A. Reynolds and T. F. Quatieri, *Approaches to Speaker Detection and Tracking in Conversational Speech*, Digital Signal Processing, 10 (2000), pp. 93-112.
[21] D. A. Reynolds and P. A. Torres- Carrasquillo "Approaches and Applications of Audio Diarization," ICASSP 2005
[22] S. E. Tranter and D. A. Reynolds, *An Overview of Automatic Speaker Diarisation Systems*. IEEE Transactions on Audio, Speech and Language, Special Issue on Progress in Rich Transcription Processing, Sept. 2006
[23] R. P. Lippmann, L. C. Kukolich and E. Singer, *LNKnet: Neural Network, Machine-Learning, and Statistical Software for Pattern Classification*, Lincoln Laboratory Journal, 6 (1993), pp. 249-268.
[24] W. M. Campbell, D. A. Reynolds, J. P. Campbell, and K. J. Brady, "Estimating And Evaluating Confidence For Forensic Speaker Recognition," ICASSP 2005