QUT/IBM
NIST 2006 SRE
System Description

Robbie Vogt, Brendan Baker, and Sridha Sridharan
Speech and Audio Research Laboratory, Queensland University of Technology
GPO Box 2434, Brisbane, AUSTRALIA, 4001.
Contacts: {r.vogt, bj.baker}@qut.edu.au

Jason Pelecanos, Jiri Navratil
IBM T.J. Watson Research Center,
Yorktown Heights, NY 10598, USA
Contacts: {jiri,jwpeleca}@us.ibm.com

I. Introduction

A. System overview

The QUT primary system uses the output from 8 different systems, with output fusion used to determine an overall score and decision. The 8 main systems are are:

1) Acoustic GMM-UBM system
2) Text-constrained Acoustic HMM using a Syllable Length Framework
3) Lexical/Idiolect system
4) Phonetic Ngram system (OGI multilingual corpus trained phone streams)
5) Phonetic Ngram system (Callhome trained phone streams)
6) Prosodic Gesture System
7) Channel/Session Variability SVM system *
8) Decision Tree System * (Phone Streams and GMM-Nbest Index)

** Designates modelling/scoring provided by IBM*

Results for this primary system are designated QUT_1 in the submission. It should be noted for the 3conv4w and 8conv4w conditions, the text-constrained HMM system was not utilised. Due to an error in the the original submission, the SVM system was also excluded from the QUT_1 system. It was, however, included in a late submission, designated as QUT_3. A secondary set of results was also submitted. These results were obtained using only the score from the acoustic GMM-UBM system. This secondary system is designated QUT_2

B. Evaluation tasks performed

Three evaluation tasks were addressed with both the primary and secondary QUT systems. All conditions tested only a single conversation side (1conv4w) while each varied in the length of training data - one side (1conv4w), three sides (3conv4w) or eight sides (8conv4w).

C. Development data and procedures

During development of the QUT submission, the NIST2005 SRE evaluation data was used as the development database and testing protocol. Data from the NIST2004 and Switchboard II protocols were utilised for background model training, and score normalisation.

II. Acoustic GMM-UBM system

The acoustic subsystem was based on the GMM-UBM approach pioneered by Reynolds [1] with the addition of a channel/session variability compensation scheme based on the approach of Kenny, et al. [2] and as described in [3] and [4].

A. Feature extraction

The acoustic GMM-UBM system used short-term cepstral-based feature vectors consisting of 12 MFCCs and 12 corresponding delta coefficients. Before the features were extracted, the audio was band filtered between 300Hz and 3.2KHz, followed by an energy based speech activity detection (SAD) process. After the features had been extracted, feature warping [5] was also applied using a 500 frame window.

B. Channel Compensation

A model-space channel compensation scheme was used in this system based on the work of Kenny, et al. [2] and further described in a submitted paper [3]. This scheme incorporates an offset to the mean vectors of a speaker GMM for *each* segment to represent session variability with this offset confined to a low-dimensional subspace (the channel subspace). This subspace was trained on a combination of NIST 2004 and Switchboard II data using a EM procedure similar to [6]; a 50-dimensional subspace was used for this evaluation. In contrast to [2], we did not incorporate a speaker subspace.

C. Model Training

Gender dependent acoustic UBMs were trained based on all the NIST 2004 data with a selection of Switchboard II, Phase 2 and 3 data to increase the diversity of speakers represented (approximately 1.5 million feature vectors per gender after a 1:3 sub-sampling). 512-component GMMs were used throughout.

Individual speaker models were adapted from the UBMs using a MAP process that simultaneously optimised both the speaker mean and a channel subspace vector for each training session. For efficiency/practicality, this simultaneous optimisation used an approach similar to the Gauss-Seidel method for solving linear systems:

1) Initialise all the session vectors and speaker model supervector estimates to $\mathbf{0}$.
2) Calculate the statistics and component occupancies of the observations in each training session based on the current variable estimates.
3) Re-estimate the session vector $\boldsymbol{x}_i(s)$ for each training session based on these statistics and the current estimate of the speaker supervector $\boldsymbol{m}(s)$.
4) Re-estimate the speaker model supervector $\boldsymbol{m}(s)$ based on these statistics and the new estimate of the session vectors $\boldsymbol{x}_i(s)$ obtained in step 3.

In contrast to the 2005 submission, only a single iteration of this optimisation was used this year.

D. Scoring and Normalisation

Scoring was performed using a version of Top-N ELLR scoring modified to incorporate channel compensation. Effectively, a channel subspace vector was estimated for each test segment/speaker model combination, in a similar fashion to the training procedure, before the ELLR score was calculated. For efficiency, the channel vector estimation used statistics calculated on the UBM so that the test segment was only processed in one additional pass to normal ELLR scoring.

Two forms of score normalisation (T-Norm and Z-Norm) were utilised based on Auckenthaler et al. [7]. T-Norm models were trained on NIST 2004 data, with approximately 40 female and 27 male models from each split for each training length (giving around 120 female and 80 male T-Norm models for running the evaluation). Z-Norm segments also came from NIST 2004 data using a distinct set of speakers to the T-Norm models. Approximately 130 segments per gender were used for the evaluation.

E. Processing Time Estimates

- 1 Side Training – 25s/speaker
- 3 Sides Training – 77s/speaker
- 8 Sides Training – 208s/speaker
- 1, 3 and 8 Side Testing (combined) – 1041m total (not including Z- and T-Norm)

III. Text-constrained Acoustic HMM using a Syllable-length Framework

This system uses a syllable-length framework to recognise and segment the speech. Hidden Markov models are then used to model each of the syllabic events for a particular speaker. The framework was originally developed for a language identification task [8], but has since been adapted for use in speaker recognition [9].

A. Syllabic Segmentation

The syllabic segmentation was achieved by recognising broad phonetic classes (BPC) using a multilingual broad phone recogniser, and then concatenating these phones into triplets to form pseudo-syllabic events. Six broad phonetic classes are defined, resulting in a pseudo-syllabic set size of 216. In time stamping each instance of the syllables, overlapping windows were used. The broad phone recogniser was trained using the Callhome and OGI multilingual corpora. Further details on the phone recogniser, broad class descriptions and the syllabic segmentation can be found in [8].

B. Feature Extraction

After the syllabic segmentation, the boundary information was used to extract features and train individual classifiers for each syllabic event. Feature vectors, calculated every 10ms, consisting of the first 12 MFCCs and their corresponding delta coefficients were used (resulting in a feature vector length of 24). Before the features were extracted, the audio was band filtered between 300Hz and 3.2KHz, followed by an energy based speech activity detection (SAD) process. After the features had been extracted, feature warping was also applied [5].

C. Model Generation

Hidden Markov model generation and scoring was performed using HTK. A seven state left-to-right HMM topology was used. 16 mixture components were used to model each emitting state's distribution.

Background HMMs were trained for each syllable using the NIST2004 SRE evaluation data. A maximum of 50000 training instances were used for each syllable in the training of these background models. For each target speaker, the 216 syllable HMM models were then adapted from the UBM using MAP adaptation (only a single iteration of MAP was performed). In the case that no data was available for adaptation for a particular syllable, the background model was used.

D. Scoring and Optimisation

Scoring for each syllable was performed separately, resulting in 216 sets of classifications. Each syllable classifier produced its own set of ELLR scores. T-Norm was performed on the ELLR scores using 100 male and 100 female models from the NIST2004 corpora. The scores from each of these classifiers were fused at the output level using linear weights calculated using a linear logistic regression algorithm. The FoCal toolkit developed by Niko Brummer was used to calculate these weights.

E. Processing Time Estimates

- 1 Side Training - 45s/speaker
- 1 Side Testing - 1300min

IV. Lexical System

The lexical system aims to exploit the differences of a speakers's idiolect (personal lexicon). The QUT lexical system is based upon an n-gram analysis technique first described in [10] and makes use of the Byblos ASR transcriptions. Both traditional bigram (conditional likelihood) and bag-of-bigram (joint likelihood) modelling was used in this implementation.

Speaker specific models were adapted from gender dependent UBMs using the MAP adaptation process outlined in [11]. The UBMs were trained using the ASR transcriptions for the NIST2004 SRE evaluation data.

Two score streams were output from this system, one for each modelling technique. T-Norm and Z-Norm were applied to these scores using models and test segments from the NIST2004 corpus.

A. *Processing Time Estimates*

- Feature extraction - Unknown (Dependent on Byblos ASR system)
- 1 Side Training - $<$ 1s/speaker
- 3 Sides Training - $<$ 1s/speaker
- 8 Sides Training - $<$ 1s/speaker
- 1, 3, 8 Side Testing (combined) - 480min.

V. Phonetic N-gram System - OGI Phone Streams

The phonetic system acts to exploit the personal variations in pronunciation and an individuals' tendency to vocalise a sequence of phones in similar ways. The QUT phonetic speaker recognition system is based on the gender-dependent phonetic refraction technique described in [12].

A. *Phone Recognition*

The phonetic system used phone transcriptions obtained from multiple open-loop phone recognisers (OLPR) each trained on one of 6 languages; English, German, Hindi, Japanese, Mandarin and Spanish. The multi-stream decoding was performed using QUT's HMM based OLPR - trained on the OGI multi-lingual database.

B. *Modelling and Scoring*

Background models were trained by gathering bi-gram and bag of 3gram statistics from the OLPR transcriptions of the NIST2004 SRE evaluation data. In order to combat model sparsity issues, MAP adaptation was used to adapt speaker models from these well trained background models [11].

ELLR scores were calculated for each modelling technique, for each stream. These scores were then normalised using Z-Norm and T-Norm segments from the NIST2004 corpus. For each modelling type (bi-gram or bag-of-3gram) the scores from the various streams were fused using linear weights calculated through a logistic regression algorithm.

C. *Processing Time Estimates*

- Feature extraction - 2s/language/file
- 1 Side Training - $<$ 1s/speaker
- 3 Sides Training - $<$ 1s/speaker
- 8 Sides Training - $<$ 1s/speaker
- 1, 3, 8 Side Testing (combined) - 400min.

VI. Phonetic N-gram System - Callhome Phone Streams

This system used identical modelling/scoring as the OGI phone stream system above, but was performed on additional set of OLPR streams provided by phone recognisers trained on the Callhome and Switchboard corpora. Six open-loop phone recognisers were used. English, German, Japanese, Mandarin, Spanish and a Multilingual phone set. Due to the increased training data available from the Callhome corpus, the HMMs for decoding were able to be trained using a significantly higher number of mixture components. This did, however, also result in longer decoding time.

A. *Processing Time Estimates*

- Feature extraction - 20s/language/file
- 1 Side Training - $<$ 1s/speaker
- 3 Sides Training - $<$ 1s/speaker
- 8 Sides Training - $<$ 1s/speaker
- 1, 3, 8 Side Testing (combined) - 400min.

VII. Prosodic Gesture System

Based on the work of Adami et al [13], this system converts f0 and energy contours into descriptive tokens representing piecewise linear segments for subsequent modelling and scoring using n-gram techniques.

A. Tokenisation

To obtain the piecewise linear tokens, the following steps were followed. Firstly, the f0 and short-time energy values were computed every 10ms. The f0 values were obtained using the getf0 utility [14], which implements a pitch tracking algorithm based on the use of the cross correlation function and dynamic programming. The time derivatives for both the f0 and energy trajectories were then calculated by fitting a straight line to 5 frames (100ms). The speech was then segmented using the zero-crossings of this rate of change values and broad phone boundaries extracted from a broad phone recogniser. Segments were then labelled according to the slope of the two trajectories, the current phonetic category, and the length of the segment. Unvoiced segments with duration shorter than 12 frames were labelled as ”Short”, and all others labelled ”Long”. For voiced segments, durations shorter than 6 frames were labelled as ”Short”, and all others labelled ”Long”.

B. Modelling and Scoring

N-gram models, were used to produce speaker dependent models. Background models were trained by gathering bi-gram and bag of 3gram statistics from the prosodic token sequences of the NIST2004 SRE evaluation data. In order to combat model sparsity issues, MAP adaptation was used to adapt speaker models from these well trained background models [11].

ELLR scores were calculated for each modelling technique, for each stream. These scores were then normalised using C-Norm and T-Norm segments from the NIST2004 corpus. C-Norm labels were obtained using the IBM channel recogniser [15].

C. Processing Time Estimates

- Feature extraction - 7s/file
- 1 Side Training - $<$ 1s/speaker
- 3 Sides Training - $<$ 1s/speaker
- 8 Sides Training - $<$ 1s/speaker
- 1, 3, 8 Side Testing (combined) - 400min.

VIII. Support Vector Machine (SVM) system - IBM

The SVM component system is based on the use of the GLDS kernel [16]. Here, the GLDS kernel feature space is created from Gaussian Mixture Models (GMMs) determined through maximum-a-posteriori (MAP) adaptation [1] of a Universal Background Model (UBM) to the features of a speaker’s utterance. The mixture component means are adapted while the weights and variances remain constant. The MAP adaptation of the mixture component means is performed according to an inter-session variation (ISV) model constraint [4], [17]. The feature space of the SVM is based on the supervector formed from the concatenation of the adapted mixture component mean vectors. More specifically, the SVM feature space is established by taking the difference between the supervector of the concatenated Gaussian means of Universal Background Model (UBM) from the supervector formed from the means of the adapted GMM. Each dimension of the SVM feature space is further normalized by the standard deviation of a set of development set impostor models (derived from NIST 2004 data).

The process for training a speaker model consists of creating a target speaker supervector from a training utterance and a set of impostor supervectors from a group of out-of-set speaker utterances. An optimal separating hyperplane is then calculated. The speaker model is represented by a normal vector to the hyperplane and an offset.

In testing, the test utterance is used to adapt a UBM and is normalized to form a single variance normalized SVM feature space vector in the normal manner. A dot product between this vector and the normal to the training model hyperplane with the offset is used to provide the speaker score. This score was further normalized using Z-Norm and T-Norm [7] (constructed from NIST 2004 audio). In addition to these techniques, symmetric scoring was also performed (see [15]). In this forward scoring scenario, the trials are first scored in the normal manner. In the reversed scoring scenario, the test utterances and training utterances are switched such that models are created from the test utterances while these models are scored using the training utterances.

IX. Binary-decision Tree system - IBM

Phonetic, lexical, and other decoders can be used as speech ”tokenizers” to transform the raw signal into a sequence of discrete units. These, in turn, are analyzed using models with a binary-decision tree (BT) structure to exploit information encoded by the inherent time dependencies. In general, a 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 have 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 system submission, 12 phonetic decoders (as described above) and one GMM tokenizer were used to generate sequences of tokens with an inventory defined as the phone repertory of each corresponding recognizer, and as a set of indices $\{0, .., 511\}$ for the GMM-index (GIX) tokenizer respectively.

A. BT Training

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 [18].

First, on data from background speakers (NIST 2004 SRE), a common BT model was created resulting in a BT with about 3k terminal nodes using up to 3 predictors (i.e. exploiting a context of 4 tokens at a time). Subsequently, individual target speaker models were created using an adaptive BT training algorithm from the common BT model as described in [19]. Three variants of the GIX system were trained from index sequences by leaving out 0, 1, and 2 frames (subsampling).

B. BT Scoring and normalization

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 [19]. 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 [1] - 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 [15]). 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)

The final 12 phonetic and 3 GIX out scores were fed into the fusion back-end.

C. Processing Time Estimates

Excludes ASR transcription and file IO loading models into memory:

- Processor class: Xeon CPU 2.8GHz Dual, 2GB RAM, Linux Fedora Core 1
- 1 Target enrollment $< 10ms$
- 1 Trial: 2ms (phonetic BTs), 5ms (GIX BTs)

X. Overall Fusion

In order to obtain an overall score for each test segment, the scores from the various subsystems were fused. Fusion was performed on the output scores using linear weights calculated through use of a logistic regression algorithm. This was performed using the FoCal toolkit provided by Niko Brummer. As well as the subsystem scores, meta data was also used as input into the fusion process. Inputs representing the channel type (detected by the IBM channel recogniser) and the model gender made up this meta data.

References

[1] D. Reynolds, “Comparison of background normalization methods for text-independent speaker verification,” in *Eurospeech*, vol. 2, 1997, pp. 963–966.

[2] P. Kenny and P. Dumouchel, “Experiments in speaker verification using factor analysis likelihood ratios,” in *Odyssey: The Speaker and Language Recognition Workshop*, 2004, pp. 219–226.

[3] R. Vogt, B. Baker, and S. Sridharan, “Modelling session variability in text-independent speaker verification,” in *Interspeech - Eurospeech*, 2005.

[4] R. Vogt and S. Sridharan, “Experiments in session variability for modelling for speaker verification,” in *ICASSP*, 2006.

[5] J. Pelecanos and S. Sridharan, “Feature warping for robust speaker verification,” in *A Speaker Odyssey, The Speaker Recognition Workshop*, 2001, pp. 213–218.

[6] P. Kenny, G. Boulianne, and P. Dumouchel, “Eigenvoice modeling with sparse training data,” *IEEE Trans. on Speech and Audio Processing*, in press.

[7] R. Auckenthaler, M. Carey, and H. Lloyd-Thomas, “Score normalization for text-independent speaker verification systems,” *Digital Signal Processing*, vol. 10, no. 1/2/3, pp. 42–54, 2000.

[8] T. Martin, B. Baker, E. Wong, and S. Sridharan, “A syllable-length framework for language identification,” *Computer Speech and Language*, 2006.

[9] B. Baker, R. Vogt, and S. Sridharan, “Gaussian mixture modelling of broad phonetic and syllabic events for text-independent speaker verification,” in *Interspeech - Eurospeech*, Portugal, 2005.

[10] G. Doddington, “Speaker recognition based on idiolectal differences between speakers,” in *Eurospeech*, vol. 4, Denmark, 2001, pp. 2517–2520.

[11] B. Baker, R. Vogt, M. Mason, and S. Sridharan, “Improved phonetic and lexical speaker recognition through MAP adaptation,” in *Odyssey: The Speaker and Language Recognition Workshop*, 2004, pp. 94–99.

[12] W. Andrews, M. Kohler, J. Campbell, and J. Godfrey, “Phonetic, idiolectal, and acoustic speaker recognition,” in *A Speaker Odyssey, The Speaker Recognition Workshop*, 2001.

[13] A. Adami, R. Mihaescu, D. Reynolds, and J. Godfrey, “Segmentation for Speaker and Language Recognition,” in *Proc. European Conference on Speech Communication and Technology (Eurospeech)*, Geneva, 2003.

[14] D. Talkin, *Speech Coding and Synthesis*. New York: Elsevier, 1995, ch. A Robust Algorithm for Pitch Tracking (RAPT).

[15] J. Pelecanos, J. Navratil, M. Omar, and G. Ramaswamy, “The ibm nist sre05 system description,” in *NIST SRE05 Workshop*, 2005.

[16] W. Campbell, “Generalized linear discriminant sequence kernels for speaker recognition,” in *ICASSP*, vol. 1, 2002, pp. 161–164.

[17] P. Kenny, G. Boulianne, P. Oullet, and P. Dumouchel, “Improvements in factor analysis based speaker verification,” in *ICASSP*, 2006.

[18] J. Navratil, “Recent advances in phonotactic language recognition using binary decision trees,” in *Interspeech*, 2006,submitted.

[19] J. Navratil, Q. Jin, W. Andrews, and J. Campbell, “Phonetic speaker recognition using maximum likelihood binary-decision tree models,” in *ICASSP*, Hong Kong, 2003.

[20] T. Joachims, “Text categorization with support vector machines: Learning with many relevant features,” in *Proc. of European Conference on Machine Learning*, 1998.