ABC NIST SRE 2016 SYSTEM DESCRIPTION

Niko Brummer[1], Albert Swart[1], Jesús Jorrín-Prieto[1], Paola García[1], Luis Buera[1], Pavel Matějka[2], Oldřich Plchot[2], Mireia Diez[2], Anna Silnova[2], Xiaowei Jiang[2], Ondřej Novotný[2], Johan Rohdin[2], Hossein Zeinali[2], Ondřej Glembek[2], František Grézl[2], Lukáš Burget[2], Lucas Ondel[2], Jan Pešan[2], Jan "Honza" Černocký[2], Patrick Kenny[3], Jahangir Alam[3] and Gautam Bhattacharya[3]

[1]Agnitio, Voice ID.
{nbrummer,aswart,jjorrin,pgarcia,lbuera}@agnitio-corp.com
[2]Brno University of Technology, Speech@FIT and IT4I Center of Excellence, Brno, Czech Republic
{matejkap,iplchot,cernocky,inovoton}@fit.vutbr.cz
[3]CRIM, Montreal (Quebec), Canada
{patrick.kenny,jahangir.alam,gautam.bhattacharya}@crim.ca

Index Terms— automatic speaker identification, deep neural networks, bottleneck features, i-vector, PLDA, snorm

1. INTRODUCTION

This submission is a collaborative/competitive effort of Agnitio, BUT and CRIM.

2. AGNITIO

Agnitio's final system is based on three subsystems: MFCC-PLDA, MFCC-BNF-4-PLDA and MFCC-BNF-2-PLDA.

2.1. Datasets

We employed SRE 2004-2008 and Fisher data to train our system. The datasets are as follows:

- SRE04, SRE05, SRE06 and SRE08 data were used to train the UBM, i-vector extractor, NDA and the PLDA,
- 300 hours of Fisher database was used to train a DNN,
- Unlabelled development data was used for mean and score normalization.

2.2. Feature Extraction

We used two feature extractors: MFCCs and Bottleneck DNN features.

Xiaowei Jiang's stay and work at BUT was partly supported by the SpeechLab, Department of Computer Science and Engineering, Shanghai Jiao Tong University, Shanghai, China.

This work was supported by the DARPA RATS Program under Contract No. HR0011-15-C-0038. The views expressed are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

This work was also supported by the Intelligence Advanced Research Projects Activity (IARPA) via Department of Defense US Army Research Laboratory contract number W911NF-12-C-0013. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright annotation thereon. Disclaimer: The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of IARPA, DoD/ARL, or the U.S. Government.

The work was also supported by Czech Ministry of Interior project No. VI20152020025 "DRAPAK" and European Union's Horizon 2020 programme under grant agreement No. 645523 BISON.

This project has received funding from the European union's Horizon 2020 research and inovation programme under the Marie Sklodowska-Curie and it is co-finanaced by the South Moravian Region under grant agreement No. 665860.

- We extracted 20 MFCC static coefficients (C0-C19) from 250Hz to 3.400Hz including delta and delta-delta. The analysis window is 20 ms long with a frame rate of 10ms. Two combined systems: Long-Term Spectral Divergence (LTSD) VAD and energy based VAD computed speech/nonspeech labels. First 50ms of every audio are removed to avoid inconsistent VAD behavior. Once MFCC coefficients are computed, they are normalized applying Cepstral Mean Normalization (CMN), RelAtive SpecTral Amplitude (RASTA) processing and warping (3 seconds window).
- BNF features were computed using a 5-hidden layer DNN, trained using 300 hours of data from Fisher database. Each BNF is 60 dimensional. Pnorm was selected as the appropriate the non-linearity. It has 500 maxout units with inputs of dimension 5000. Spliced MFCCs (with a context of 4 frames to the left and 4 frames to the right) were used as the input for the DNN.

2.3. Classifier Schemes

MFCC-PLDA: Our basic system is based on a full covariance Universal Background Model (UBM) of 2048 GMM-component, using MFCCs in the whole process. 400 dimensional i-vectors are extracted consequently. Nearest-neighbor Analysis (NDA) performs a dimensionality reduction of those ivectors from 400 to 250. This process is followed by mean normalization, which is adapted to the use case employing unlabeled development data, and length normalization. Scoring between i-vectors is achieved by using gender dependent PLDA (speaker space dimension is fixed to 120).

AGN-MFCC-BNF-4-PLDA: Two feature extractors are used: MFCC and BottleNeck Features (BNF). The bottleneck postion is in the fourth-hidden layer.

Two full covariance Universal Background Models (UBMs) composed of 2048 component are trained on MFCC and BNF features. For each audio, two 400-dimensional i-vectors are extracted, respectively. They are then stacked to obtain a single 800-dimension i-vector per audio. Once again, NDA is employe, but this time it performs a dimensionality reduction of those ivectors from 800 to 500. The process is followed by mean normalization, and length normalization. Scoring between i-vectors is achieved using gender dependent PLDA (speaker space dimension fixed to 200).

AGN-MFCC-BNF-2-PLDA: MFCC-BNF-2-PLDA is identical to MFCC-BNF-4-PLDA, but modifying the position of the bottleneck. In this case it is allocated in the second hidden layer.

2.4. Normalization

For each classifier we employed gender dependent s-norm. Score normalization cohorts are adapted to the use case, using SRE16 unlabeled development. Automatic gender identification is obtained from i-vectors in PLDA framework.

2.5. Fusion

Normalized scores for the three subsystems are linearly fused by a simple weighted addition. Weights are 0.5, 0.25 and 0.25 for MFCC-PLDA, MFCC-BNF-4-PLDA and MFCC-BNF-2-PLDA, respectively. The scores are assumed to be of comparable scales because of score normalization.

2.6. Performance and Processing Requirements

The infrastructure used to run the exprimentes is a CPU, Intel(R) Core(TM) i7-2600 CPU @ 3.40GHz, with a total memory of 8140568kB.

The execution time of an enrollment model in a single thread is of 13.82RT, using 2.97GB of memory. On the other hand, each trial employs 13.92RT and 2.97GB of memory to calculate a score.

3. BUT

All BUT systems are based on ivector paradigm [1] with different features and backend.

3.1. DATASET

- Primary Background Data: telephone data from NIST SRE 2004 - 2008, Fisher English and Switchboard
- nonEnglish data: we selected non English segments from our Primary Background data. We split this data into 3 parts- **train** (for PLDA, LDA, SVM, SNORM ...), **dev** (calibration, fusion), **test** (blind test set). We followed the split between training and development data designed in the PRISM dataset and we also created short cuts with durations of speech which reflects the evaluation plan for NIST SRE 2016 - more precisely we based our cuts on the actual detected speech in the NIST SRE 2016 development labeled data. We chose the cuts to follow the uniform distribution:
  - Enrollment between 25-50 sec of speech
  - Test between 3-40 sec of speech
  - Train between 10-60 sec of speech

- Unlabeled data: unlabeled data from SRE16 development

3.2. VAD

Our VAD is based on phone recognizer trained on Fisher with 3 variants of Fisher with added noise at different SNR. We dropped all frames that were marked as silence or noise.

3.3. Features

All features have 19 direct coefficients with Energy or C0, delta and double delta coefficients, which makes it 60 dimensional features. Short time cepstral mean and variance normalization over 3 second window (sCMVN) is applied only on speech frames. We used 3 sets of features:

- 19 MFCC+Energy - HTK based
- PLP+Energy - HTK based
- Perseus - description of these features can be found in[2].
- MFCC+SBN-ENG (Stacked Bottleneck Features) trained on provided English data
- MFCC+SBN-BABEL (Stacked Bottleneck Features) trained on provided BABEL data

3.4. UBM+iVector

We used GMM with 2048 Gausians and iVector with 600 dimensions trained in gender independent fashion. Both components are trained only on the telephone data from MIXER collection, Fisher English and Switchboard 2. UBM is trained on random selection of 8000 files. iVector is trained on the 74594 files from 16241 speakers.

3.5. Classifier

We have 3 variants of classifiers:

- **PLDA:** For training PLDA model, telephone data from Primary Background Data and Non English data were used. All iVectors were mean (mean was calculated using all training data) and length normalized, followed by LDA with Within Class Covariance Correction (WCC) decreasing dimensions of vectors from 600 to 200. The WCC is based on weighted adding of the within-class covariance of different languages and datasets into the within-class covariance of LDA. We were also removing shift between the training data and the minor and major datasets. Resulting scores were normalized using speaker dependent snorm with cohort from Primary Backround data and unlabeled data. Speaker dependent means for the snorm were computed on 500 closest ivectors for each speaker.
- **Discriminative PLDA:** For training DPLDA model, telephone data from Mixer+Fisher+Switchboard was used along with unlabeled data from NIST SRE16. Unlabeled data were used to form non-target trials with labeled telephone data only (e.g. no trials between two unlabeled utterances were used for training). First, NAP was performed on top of all ivectors. As classes for NAP, 20 languages from training list were selected along with one class corresponding to both major and minor unlabeled data. After NAP all ivectors were mean (mean was calculated using all training data available) and length normalized. After the mean normalization, we performed LDA, decreasing the dimensionality of vectors to 250. As an initialization of DPLDA training, we used a corresponding PLDA model. During the DPLDA training, we set the prior probability of target trials to reflect the SRE16 evaluation operating point.
- **Support Vector Machines:** One SVM per speaker was trained using the enrollment ivector(s) as possitive samples and unlabeled major and unlabeled minor data as negative samples. Length normalization, WCCN and NAP were applied to ivectors. All these were trained with telephone data from Mixer+Fisher+Switchboard and the classes for NAP were languages present in the training data[1]. ZTNorm was applied to system scores. ZNorm was trained on a subset of Chinese utterances from the training portion of non-English short cuts, plus the data from unlabeled major and unlabeled minor sets. TNorm was trained with the SVM models trained on Chinese cuts, using the unlabeled major an minor sets as background data (negative samples).

3.6. Fusion

Fusion and calibration of the BUT subsystems were trained with logistic regression optimizing the cross-entropy between the hypothesized and true labels on our development set composed of non-English short segments. Our objective was to improve the error rate on

[1]List of languages used for NAP: ['USE', 'ENG', 'CHN', 'RUS', 'ARB', 'YUH', 'THA', 'SPA', 'VIE', 'HIN', 'JPN', 'BEN', 'KOR', 'WUU', 'TGL', 'FAR', 'CFR', 'CHN.YUH', 'CHN.WUU', 'ITA']

the test part of our Non English test set, but we were also monitoring error-rate trends on the labeled minor SRE'16 development set.

The following subsystems were used in the final fusion:

- 2 DPLDA systems trained on PLP and MFCC,
- 4 PLDA systems trained on MFCC, Perseus, PLP, MFCC-SBN
- SVM system trained on PLP

We produced two different fusions that we denote as BUT-GI-BIG2 and BUT-GI-BIG3. The only difference between these fusions is in a single system - PLDA system trained on MFCC-SBN. The SBNs for fixed condition were trained on English data producing the BUT-GI-BIG3, while for the open condition, we used BABEL languages. We denote the fusion for the open condition as BUT-GI-BIG2.

Each subsystem was pre-calibrated by the logistic regression and then all subsystems were fused by the means of logistic regression using the development set from our Non English development set that contains only the short cuts. We report the results of our fusions and all individual subsystems in the Table 1.

3.7. Performance and Processing Requirements

The infrastructure used to run the expriments is a CPU, Intel(R) Xeon(R) CPU 5675 @ 3.07GHz, with a total memory of 37GB.

The execution time of iVector extraction process in a single thread is of 18 times faster than real time (FRT) (computed only on detected speech, would be 41FRT computed for whole recordings including silence) for MFCC only system and 3.3 FRT for the MFCC-SBN system, using 3GB and 5GB of memory respectively. Enrollment and scoring is negligible with respect to the iVector extraction time for all our backends.

4. CRIM

We developed speaker verification systems based on three different speech front-ends: MFCC, LFCC and LPCC. For scoring we made use of 3 classifiers - cosine distance (CD), Probabilistic Linear Discriminant Analysis (PLDA) and Latent Dirichlet Allocation (LDA). CRIM's final system combines 8 sub-systems : LFCC-CD, LFCC-PLDA, MFCC-CD, MFCC-DNN-LDA, MFCC-DNN1, MFCC-DNN2, MFCC-DNN3 and LPCC-CD.

4.1. Datasets

We used SRE 2004-2008 and Switchboard data as background data for our systems. The dataset was partitioned into two parts:

- Oriental Backgroud Data: This set includes recordings from Chinese, Mandarin and Tagalog (from SRE 2004-2008 data).
- Primary Background Data: Everything else
- Oriental Data: We refer to the combination of the Oriental Background Data and the SRE 2016 unlabelled data as the Oriental Data.

4.2. Pre-processing

VAD: We removed all non-speech frames using an unsupervised GMM-based voice activity detector.

Feature Extraction: We extracted MFCC, LFCC and LPCC features from all the recordings. All features are 60 dimensional. These incluse 20 static coefficients including log energy, 20 delta and 20 delta-delta coefficients. Short-term mean and variance normalization is also performed.

UBM: We trained a 2048-component diagonal covariance UBM using the primary background data. This is then iteratively adapted to the Oriental data using relevence MAP.

4.3. i-vector Extractor

First we trained an i-vector extractor using sufficient statistics extracted from all of the primary background data. We use this model as an initialization we performed several iterations of minimum divergence training on the Oriental data. This extractor is used for extracting the i-vectors used in all our systems.

4.4. MFCC-DNN Systems

In order to train a speaker classifier network (SCN) we use a feedforward neural network to learn mapping between i-vectors and speaker labels. Projecting the i-vectors into a higher dimension space significantly improves speaker discriminant properties of the resulting features [3]. Projecting the i-vectors into a higher dimensional space significantly improves speaker discriminative properties of the resulting features [3]. The SCN is two layers deep and uses sigmoid nonlinearity in the hidden layers. Each hidden layer consists of 2000 hidden units. The softmax output distribution is over 4323 speakers in the

background set (Primary Background Data + Oriental Background Data). The speakers are filtered based on the number of their recordings. Speakers having at least 5 recordings/i-vectors are selected. We make use of i-vectors that have been adapted to the oriental data for SCN training. The i-vectors are length normalized before being processed by the SCN. After the model is trained, it is used as a feature extractor for the background, enrolment and test data. Specifically, we extract the activations of the last hidden layer and treat them as feature vectors (d-vectors) for speaker verification. In the case of SRE16 data (development and evaluation) we only make the d-vectors to be of unit norm and do not perform any mean-centering. Speaker verification is performed using a cosine distance classifier with the SCN-projected features (i.e., d-vectors). To this end with MFCC frontend we developed three system variations:

- MFCC-DNN1: For speaker models with 3 enrolment d-vectors (2000-dimensional) we average the individual scores during cosine scoring. In all other systems, for speaker models with 3 enrolment i-vectors/d-vectors a single score is produced by averaging the i-vectors/d- vectors.
- MFCC-DNN2: NAP projection is applied to all the d-vectors produced by a SCN.
- MFCC-DNN3: In this case we reduce the dimension of the NAP projected d-vectors using a principal component analysis (PCA) technique.

4.5. MFCC-DNN-LDA System

In this system we model the hidden activations of the DNN speaker classifier using Latent Dirichlet Allocation (LDA). The system MFCC-DNN-LDA differs from MFCC-DNN1 in that we replaced the cosine distance backend with a probabilistic backend which was trained blindly on the unlabelled training data. (We did not attempt to assign speaker, language or gender labels to the training data.)

As in MFCC-DNN1, the feature vector used to represent an utterance consisted of the sigmoid activations of the last hidden layer of the DNN. We viewed these features as noisy binary vectors and modeled them by a hidden vector of Bernoulli probabilities. If speaker labels were available, we would associate one Bernoulli probability vector with each speaker. Since we did not have speaker labels for the training set, we treated the recordings as if they all came from different speakers. We treated the components of the feature vector as being statistically independent and we placed a Beta prior on each of the Bernoulli probabilities. We "estimated" the priors by appealing to the maximum likelihood II principle, using the methods in [3].

4.6. Performance and Processing Requirements

In order to report real time factor we conducted experiments on an Intel(R) Xeon(R) CPU X5650 @ 2.67GHz with a total memory of 94.5GB. The execution time for the extraction of i-vectors/d-vectors (VAD segmentation + Features extraction + extraction of Sufficient statistics + generation of i-vectors/d-vectors) + enrollment + scoring in a single thread is of 8 times faster than the real time using 3.5GB of memory. By d-vectors we refer to the features supplied by a speaker classifier network.

One of the advantages of working in low-dimensional i-vector space is that computation times for neural networks is modest, both in terms of network training and feature extraction.

Training: In order to provide timing information, we report the avarage number of training epochs and the average epoch duration over 10 training runs. Network training was carried out on a NVIDIA Titan X GPU. Models trained to convergence in 204 epochs on average and the average epoch duration was 10.62 sec.

Feature Extraction: We extract features from the trained network for 1000 i-vectors. This average duration of the recordings represented by this set of i-vectors is 17184 frames. Feature extraction is performed 10 times on both GPU and CPU, with the GPU being only marginally faster. The average feature computation time on the CPU is 0.0056 sec versus 0.0044 sec on the GPU.

4.7. Fusion of CRIM systems

For submission purposes, several CRIM systems were fused to produce a single set of scores. The data used for training the fusion parameters was the labelled minor SRE'16 development data. After training, the fusion was then applied: (i) to this same data (test-on-train) to pass as training scores for the final ABC fusion; and (ii) to the SRE'16 evaluation data, also as input to the final ABC fusion.

The subsystems included in the fusion were selected according to individual performance on the labelled minor SRE'16 development data. We did not base inclusion/exclusion decisions on EER or DCF. Instead we looked at the regularity of score histograms, DET-curves and normalized DCF curves. We decided not to judge system goodness by EER, because the EER operating point is too far from the SRE'16 DCF operating points. On the other hand, performance of systems and fusions according to the SRE'16 DCF criterion also did not play

a major role in our decision making, because we believe that the size of the labelled data did not permit accurate estimates of error-rates at these operating points. Indeed, we saw both individual systems and fusions could have as few as zero false-accept errors on this small database. We looked instead at DCF curves, which cover all operating points between EER and the DCF'16 operating points. The following 8 CRIM systems were used in the fusion: LFCC-CD, LFCC-PLDA, MFCC-CD, MFCC-DNN-LDA, MFCC-DNN1, MFCC-DNN2, LPCC-CD, MFCC-DNN3.

Because of data scarcity and to combat over-training, we used generative fusion and calibration strategies, with as few as possible parameters. The fusion strategy was linear-Gaussian pre-calibration of each sub-system, followed by equal-weighted summation. Separate gender-independent calibrations were done for 1-call and 3-call enrollment. The linear-Gaussian calibration is done by computing the log-LR obtained from a generative model with two univariate Gaussians for targets and non-targets, with different means and shared covariance. The parameters were estimated with maximum-likelihood. Pre-calibration was applied before summation, so that: (i) missing scores—because of VAD failure—could be replaced by log-LR = 0; and (ii) sub-system scores were roughly at the same scale, with better system contributing a bit more than weaker systems.

5. FINAL ABC FUSION & SUBMISSION

5.1. Primary ABC fusion for fixed condition

The input to the final ABC fusion consisted of 3 sets of scores, each produced by the labs Agnitio, BUT and CRIM. As explained in previous sections each of these inputs were themselves fused from multiple subsystems. Each lab provided both training scores and evaluation scores as input to the fusion. The training scores consisted of SRE'16 minor labelled development data. In the case of CRIM, this constituted a second use of this data. For the Agnitio and BUT systems, this data was unexposed.

Here also, as already explained above for the CRIM fusion, we did not judge fusion strategies by EER or DCF'16. Instead we looked at regularity of score histograms, DET-curves and normalized DCF curves. We tried two fusion strategies, both generative:

1. Independent pre-calibrations of sub-systems (linear-Gaussian), followed by combination by plain summation (no trainable parameters), followed by post-calibration (generative, non-linear).
2. Linear-Gaussian generative fusion,[2] followed by non-linear post-calibration.

We chose strategy 2: It is more powerful and therefore more risky w.r.t. overtraining, but it gave a significantly better DET-curve at all operating points.

For post-calibration after fusion, we tried linear, quadratic and NIG [4]. In all cases NIG gave much better calibration as judged on the SRE'16 minor labelled development data. The linear-Gaussian calibration is the same as described in section 4.7 above. The quadratic fusion is also generative Gaussian, but with independent (rather than shared) variances for targets and non-targets. The NIG calibration used independent normal-inverse Gaussian (NIG) distributions for targets and non-targets.

NIG parameter estimation is tricky. For NIG maximum-likelihood parameter estimation in [4], we had used a trust-region Newton algorithm for direct optimization of the likelihood. This time, we used a modified version of the EM algorithm in [5]. The modification is similar to the minimum-divergence trick—the model is over-parametrized during the M-step and then simplified again using a reparametrization of the hidden variable. The EM algorithm was initialized with moment matching. After a few hundred EM iterations, training was completed using direct L-BFGS optimization, which gives faster convergence during the end-game.

The inputs that were used in this fusion were the Agnitio fusion of section 2.5, the CRIM fusion of section 4.7 and the BUT fusion known as BUT-GI-BIG3 of section 3.6.

5.2. Primary ABC fusion for open condition

For the primary submission to the open condition, the same methods were used (linear-Gaussian fusion, followed by NIG post-calibration). The same Agnitio and CRIM systems were used, but the BUT system was BUT-GI-BIG2.

5.3. Contrastive ABC systems

For the contrastive submission 1 to both conditions we used BUT-GI-BIG3 for fixed and BUT-GI-BIG2 for open condition. The motivation for this was that these fusions did not see any of the labeled SRE16 data. For the contrastive submission 2 to fixed condition we submitted single DPLDA system trained on PLP, but we have added

[2] Here we also pre-calibrated, but this has no effect. The fusion algorithm that we applied was actually slightly non-linear, with multivariate t-distributions instead of Gaussians, where the t-distributions resulted from Bayesian marginalization over model parameters.

SRE'16 minor labeled development data to the training of DPLDA.

6. REFERENCES

[1] N. Dehak, P. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet, "Front-end factor analysis for speaker verification," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. PP, no. 99, 2010.

[2] Ondřej Glembek, Pavel Matějka, Oldřich Plchot, Jan Pešán, Lukáš Burget, and Petr Schwarz, "Migrating i-vectors between speaker recognition systems using regression neural networks," in *Proceedings of Interspeech 2015*, 2015, pp. 2327–2331.

[3] T. Minka, "Estimating a Dirichlet distribution," 2012.

[4] Niko Brummer, Albert Swart, and David van Leeuwen, "A comparison of linear and non-linear calibrations for speaker recognition," in *Odyssey 2014: The Speaker and language Recognition Workshop*, 2014.

[5] Dimitris Karlis, "An em type algorithm for maximum likelihood estimation for the normal inverse gaussian distribution," *Statistics & Probability Letters*, March 2002.

Table 1. *Comparison of systems on all labeled development data using NIST scoring tool, C_{Prm} stands for $C_{Primary}$, * results for uncalibrated scores*

Site	System Name	Features	Classifier	Condition	Use DEV16 Labeled	Equalized results			Unequalized results		
						EER[%]	$minC_{Prm}$	$actC_{Prm}$	EER[%]	$minC_{Prm}$	$actC_{Prm}$
BUT	DPLDA_PLP	PLP	DPLDA	fixed	NO	19.56	0.6482	0.8616	18.57	0.6532	0.875829
	DPLDA_MFCC	MFCC	DPLDA	fixed	NO	20.06	0.8254	0.8809	19.69	0.8201	0.876658
	PLDA_MFCC	MFCC	PLDA	fixed	NO	17.89	0.8194	1.5811	18.96	0.7995	1.489117
	PLDA_PLP	PLP	PLDA	fixed	NO	18.44	0.7136	0.8232	18.43	0.6962	0.816542
	PLDA_PERS	PERSEUS	PLDA	fixed	NO	19.62	0.8117	0.8480	18.06	0.8017	0.837174
	PLDA_MFCCSBN	MFCC+SBN-ENG	PLDA	fixed	NO	20.96	0.7917	0.8644	22.68	0.7853	0.864065
	PLDA_MFCCSBN_BABEL	MFCC+SBN-BABEL	PLDA	open	NO	16.58	0.7373	0.8320	18.66	0.7357	0.836106
	SVM_PLP	PLP	SVM	fixed	NO	18.04	0.7545	2.8636	18.28	0.7542	2.624896
AGN	MFCC-BNF-4-PLDA	MFCC+BNF	PLDA	fixed	NO	16.14	0.6515*	0.8263*	17.43	0.6577*	0.840758*
	MFCC-BNF-2-PLDA	MFCC+BNF	PLDA	fixed	NO	15.49	0.6661*	0.8384*	16.61	0.6725*	0.850437*
	MFCC-BNF-FUSION	MFCC+BNF	PLDA	fixed	NO	15.71	0.6427*	0.9304*	16.88	0.6485*	0.942005*
CRIM	LFCC-CD	LFCC	CD	fixed	NO	20.06	0.7860	0.8174	20.06	0.7860	0.8174
	LFCC-PLDA	LFCC	PLDA	fixed	NO	21.19	0.8147	0.9662	20.79	0.8154	0.976265
	MFCC-CD	MFCC	CD	fixed	NO	18.14	0.7207	0.7722	17.65	0.7067	0.751684
	MFCC-DNN-LDA	MFCC	LDA	fixed	NO	15.04	0.8428	0.8863	15.09	0.8064	0.847248
	MFCC-DNN1	MFCC	CD	fixed	NO	17.42	0.7307	0.7932	16.51	0.7154	0.784878
	MFCC-DNN2	MFCC	CD	fixed	NO	15.13	0.7455	0.8048	15.53	0.7198	0.780887
	MFCC-DNN3	MFCC	CD	fixed	NO	15.33	0.7509	0.7969	15.55	0.7189	0.771041
	LPCC-CD	LPCC	CD	fixed	NO	20.52	0.7671	0.8208	19.82	0.7620	0.803301
	Primary	-	-	fixed	YES	13.98	0.5758	0.5920	13.77	0.5363	0.548390
	Contrastive 1	-	-	fixed	NO	15.03	0.6410	0.6581	15.24	0.6172	0.640508
	Contrastive 2	PLP	DPLDA	fixed	YES	11.76	0.5475	0.5713	11.79	0.5544	0.576740
	Primary	-	-	open	YES	13.40	0.5588	0.5756	13.46	0.5122	0.534383
	Contrastive 1	-	-	open	NO	13.85	0.6368	0.6412	14.56	0.6095	0.613116