UNIVERSITY OF THE BASQUE COUNTRY (GTTS@EHU) SYSTEM FOR THE NIST 2017 LANGUAGE RECOGNITION EVALUATION

Mikel Penagarikano, Amparo Varona, Luis J. Rodríguez-Fuentes, Germán Bordel

GTTS Group (http://gtts.ehu.es), Department of Electricity and Electronics
University of the Basque Country UPV/EHU, 48940, Leioa, Spain
e-mail: mikel.penagarikano@ehu.eus

1. INTRODUCTION

This paper briefly describes the language recognition systems developed by the Software Technology Working Group (http://gtts.ehu.es) of the University of the Basque Country (EHU) for the NIST 2017 Language Recognition Evaluation. The submitted system uses the the Brno University of Technology (BUT) 80 dimension bottleneck features [1] trained on FisherEnglish (2423 triphones) and follows the Total Variability Factor Analysis (*i-Vector*) approach [2]. The *i-Vector* extractor (1024 Gaussians and 400 dimensional *i-Vector*) is based on the Sidekit Toolkit [3] and it is followed by a Gaussian Linear Classifier and a Discriminative Gaussian Backend. Linear logistic regression calibration is applied to the final scores using the FoCal Toolkit [4].

2. DATASETS

The meta-information of the audio files was not used to create the datasets. That is, each dataset contains audio files with different lengths, formats and sources. The data was partitioned as follows::

- ***Train*** (16201 files): All available data from LDC2017E22 (2017 NIST LRE Training Data). This Dataset was randomly reduced to create another dataset:
  - ***TrainBalanced*** (4566 files): Random subset of files summing around 8000 seconds of voiced feature vectors per target language.
- ***Dev*** (3659 files): All available data from LDC2017E23 (2017 NIST LRE Development Data). The audio files `lre17_ytgfvwpa.flac` and `lre17_gpupyoiu.flac` where excluded as repeated and empty/unvoiced, respectively . This dataset was randomly split into two new datasets:
  - ***Dev1*** (1829 files): First half.
  - ***Dev2*** (1830 files): Second half.

The Fisher English dataset was also used indirectly, since the BUT bottleneck extractor software was trained on it.

3. SYSTEM ARCHITECTURE

3.1. Feature extraction

Audio files where first converted to 8KHz linear PCM and then bottleneck feature vectors where extracted using the BUT bottleneck extractor software [1]. The used pre-trained NN was the so called `FisherEnglish_FBANK_HL500_SBN80_triphones2423`, a NN trained on Fisher English with 2423 senones as targets. For speech activity detection, the bottleneck extractor's internal energy based VAD was used.

3.2. I-vector Extraction

The Sidekit Toolkit [3] was used to create an ivector extractor. A gender independent 1024-mixture diagonal UBM was estimated by Maximum Likelihood, using the ***TrainBalanced*** dataset. The total variability matrix of rank 400 was estimated by 10 iterations of EM-MD on the same dataset.

3.3. Classifier

A simple generative multi-class Gaussian classifier was used to model the target languages. The distribution of language ivectors was modeled by a multivariate normal distribution $\mathcal{N}(\mu_l, \mathbf{\Sigma})$ for each target language $l \in L$, where the full covariance matrix $\mathbf{\Sigma}$ was shared across all target languages. Maximum Likelihood estimates of the language dependent means μ_l and the covariance matrix $\mathbf{\Sigma}$ were computed on the ***Train*** dataset. For each target language l, the scores of an i-vector x are given by:

$$score(x, l) = \log(N(x; \mu_l, \Sigma)) \tag{1}$$

Table 1. EHU fixed-primary system performance, on the NIST LRE 2017 *dev* set.

unequalized results			
Metric	$P_t = 0.1$	$P_t = 0.5$	Overall
minC	0.4210	0.1708	0.2959
actC	0.4182	0.1790	0.2986
EER	—	—	8.577
equalized results			
Metric	$P_t = 0.1$	$P_t = 0.5$	Overall
minC	0.5017	0.2011	0.3514
actC	0.5055	0.2091	0.3573
EER	—	—	10.394

3.4. Backend

A discriminative Gaussian pre-calibration/backend was applied to the scores. The means and the common covariance matrix where initialized with their ML estimates and then further re-estimated in order to maximize the Maximum Mutual Information (MMI) criterion.

During the development phase, the ***Dev1*** dataset was used to train the backend (***Dev2*** was used for validation), whereas for the final submission, the backend was trained on the full ***Dev*** dataset.

3.5. Calibration

Linear logistic regression calibration/fusion parameters were estimated on the development dataset (***Dev1*** during the development phase and ***Dev*** for the submission) using the FoCal Toolkit [4].

4. SYSTEM PERFORMANCE

The EHU submission consisted on a single primary system for the core *fixed* condition. The performance of this system on the NIST LRE 2017 *dev* set, using the scoring software provided by NIST is shown in Table 1.

5. PROCESSING SPEED AND MEMORY USAGE

The processing speed and memory usage was measured on a dual Xeon E5-2630v3 2.40 GHz processor, with 224 GB of RAM. Table 1 shows the processing speed and memory usage by the EHU fixed-primary system to process 1, 10 and 100 trials of 30s of speech. Comparing the speed and memory usage of 1, 10 and 100 trials is allows to detect which are the initialization requirements (i.e. the amount of time and memory required prior to process the trial). Four processing stages are reported:

Table 2. Single threaded CPU execution time (in seconds) and amount of memory used (in Megabytes) to process 1, 10 and 100 trials of 30s of speech by each processing stage of the EHU fixed-primary system.

	1xTrial		10xTrials		100xTrials	
	sec	MB	sec	MB	sec	MB
audio2bn	29	116	242	157	2420	222
bn2stat	12	146	71	203	646	410
stat2ivect	22	1111	169	1116	1449	1173
ivect2score	4	228	4	229	5	248

- *audio2bn* - Bottleneck features extraction from audio file, including audio format/rate conversion and energy based VAD estimation.
- *bn2stat* - UBM based first and second order statistics estimation.
- *stat2ivect* - iVector estimation.
- *ivect2score* - Estimation of scores.

Note that for some stages the processing time of 1, 10 or even or 100 trials is similar, while for other stages the memory footprint does not depend on the number of trials.

6. REFERENCES

[1] Radek Fer, Pavel Matejka, Frantisek Grezl, Oldrich Plchot, Karel Vesely, and Jan Honza Cernocký, "Multilingually trained bottleneck features in spoken language recognition," *Computer Speech and Language*, vol. 46, no. Supplement C, pp. 252 – 267, 2017.

[2] N. Dehak, P.J. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet, "Front-End Factor Analysis for Speaker Verification," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, no. 4, pp. 788–798, May 2011.

[3] Anthony Larcher, Kong Aik Lee, and Sylvain Meignier, "An extensible speaker identification SIDEKIT in Python," in *ICASSP*, March 2016, pp. 5095–5099.

[4] *FoCal Multi-class: Tools for evaluation, calibration and fusion of, and decision-making with, multi-class statistical pattern recognition scores*, https://sites.google.com/site/nikobrummer/focalmulticlass.

[5] Daniel Povey, Arnab Ghoshal, Gilles Boulianne, Lukas Burget, Ondrej Glembek, Nagendra Goel, Mirko Hannemann, Petr Motlicek, Yanmin Qian, Petr Schwarz, Jan Silovsky, Georg Stemmer, and Karel Vesely, "The kaldi speech recognition toolkit," in *IEEE 2011 Workshop on Automatic Speech Recognition and Understanding*. Dec. 2011, IEEE Signal Processing Society, IEEE Catalog No.: CFP11SRW-USB.

[6] Petr Schwarz, *Phoneme recognition based on long temporal context*, Ph.D. thesis, Faculty of Information Technology, Brno University of Technology, http://www.fit.vutbr.cz/, Brno, Czech Republic, 2008.

[7] Mireia Diez, "Frame-Level Features Conveying Phonetic Information for Language and Speaker Recognition," in *PhD Thesis*, University of the Basque Country, Leioa, Spain, September 2015.

[8] N. Brummer, *Generative, Fully Bayesian, Gaussian Pattern Classifier*, https://sites.google.com/site/nikobrummer/.

[9] David A Van Leeuwen and N Brummer, "Channel-dependent gmm and multi-class logistic regression models for language recognition," in *Speaker and Language Recognition Workshop, 2006. IEEE Odyssey 2006: The*. IEEE, 2006, pp. 1–8.

[10] N. Brümmer and J. du Preez, "Application-Independent Evaluation of Speaker Detection," *Computer, Speech and Language*, vol. 20, no. 2-3, pp. 230–275, April-July 2006.

[11] Yajie Miao, "Kaldi+pdnn: Building dnn-based ASR systems with kaldi and PDNN," *CoRR*, vol. abs/1401.6984, 2014.