Study of Different Backends in a State-Of-the-Art Language Recognition System

Mikel Penagarikano, Amparo Varona, Mireia Diez, Luis Javier Rodriguez-Fuentes, German Bordel

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

Abstract

State of the art language recognition systems usually add a backend prior to the linear fusion of the subsystems scores. The backend plays a dual role. When the set of languages for which models have been trained does not match the set of target languages, the backend maps the available scores to the space of target languages. On the other hand, the backend serves as a precalibration stage that adapts the space of scores. In this work, well known backends (Generative Gaussian Backend, Discriminative Gaussian Backend and Logistic Regression Backend) and newer proposals (Fully Bayesian Gaussian Backend and Gaussian Mixture Backend) are analyzed and compared. The effect of applying a T-Norm or a ZT-Norm is also analyzed. Finally the effect of discarding development signals, those with the highest scores, is also studied. Experiments have been carried out on the NIST 2009 LRE database, using a state-of-the-art Language Recognition System consisting of the fusion of five subsystems: A Linearized Eigenchannel GMM (LE-GMM) subsystem, an iVector subsystem and three phone-lattice-SVM subsystems. Best performance was attained by Gaussian Mixture Backend (1.25 EER), yielding 23% relative improvement with respect to the baseline (1.62 EER).

Index Terms: Spoken Language Recognition, Gaussian Backend, Gaussian Mixture Backend, Discriminative Gaussian Backend

1. Introduction

The general structure of a Spoken Language Recognition (SLR) system involves five stages: (1) extracting features/tokens; (2) applying a classifier which scores feature/token sequences with regard to models or target languages; (3) applying a backend to adapt/calibrate the scores; (4) doing the fusion of the scores of each subsystem; and (5) making a task dependent hard decision.

The backend plays a dual role. When the set of languages for which models have been trained does not match the set of target languages (either because non-target languages are used as *anchor* models, or because very few signals are available to train a robust target language model), the backend maps the available scores to the space of target languages. On the other hand, the backend serves as a precalibration stage that transforms the space of scores to get reliable estimates of the true class probabilitites.

Linear backends are usually applied, being the generative (ML) Gaussian backend [1] the most common approach. Nowadays, increasingly sophisticated calibration techniques are being applied and state-of-the-art systems implement adapted (MAP) Gaussian backends [2], discriminatively-trained (MMI) Gaussian backends [3] and regularized logistic regression backends [4].

In this work well known backends (Generative Gaussian Backend, Discriminative Gaussian Backend and Logistic Regression Backend) and newer proposals (Fully Bayesian Gaussian Backend and Gaussian Mixture Backend) are analyzed and compared. Score normalization techniques like the z-norm and t-norm, which reduce the variability of the likelihood scores, are also analyzed. Finally the effect of discarding development signals, those with the highest scores (that supposedly correspond to repeated speakers), is also studied. Experiments have been carried out on the NIST 2009 LRE database, using a state-of-the-art Language Recognition System consisting of the fusion of five subsystems: A Linearized Eigenchannel GMM (LE-GMM) subsystem, an iVector subsystem and three phone-lattice-SVM subsystems.

The rest of the paper is organized as follows. Section 2 briefly describes the backend and the fusion approaches. The experimental setup including the language recognition system is presented in Section 3. Results obtained in language recognition experiments on the NIST 2009 LRE database are presented and discussed in Section 4. Finally, conclusions are summarized in Section 5.

2. Backend and score fusion

Given a test signal, the set of language models outputs an score vector that is transformed by the backend. In this work, different types of backends have been compared:

- **Generative Gaussian Backend (GB)**: In a generic Gaussian backend, the distribution of language scores is modeled by a multivariate normal distribution $N(\mu_t, \Sigma)$ for each target language t, where the full covariance matrix Σ is shared across all target languages.

  Given an score vector $\mathbf{s}$ of size K, the output (calibrated) log-likelihood vector $\hat{\mathbf{s}}$ is obtained by:

  $$\hat{\mathbf{s}} = \mathbf{A}\mathbf{s} + \mathbf{b} + \mathbf{c} \quad (1)$$

  where matrix $\mathbf{A}$ contains rows $\mathbf{a}_t = \mu_t^T \Sigma^{-1}$, $b_t = -\frac{1}{2}\mu_t^T \Sigma^{-1} \mu_t$ and $\mathbf{c}$ is a constant vector (independent of the target-language and therefore, negligible) such that $c_t = -\frac{K}{2} log(2\pi) - \frac{1}{2} log|\Sigma| - \frac{1}{2}\mathbf{s}^T \Sigma^{-1}\mathbf{s}$.

  In the case of the generative Gaussian backend, Maximum Likelihood (ML) estimates of the means and the covariance matrix are usually computed.

- **Discriminative Gaussian Backend (DGB)**: The ML estimates of the means and the common covariance matrix are used initially, but further reestimates of the means are iteratively computed in order to maximize the Maximum Mutual Information (MMI) criterion:

$$F_{\mathbf{MMI}}(\lambda) = \sum_{\forall \mathbf{s}} log \frac{p_\lambda(\mathbf{s}|l(\mathbf{s}))^C}{\sum_{\forall l} p_\lambda(\mathbf{s}|l)^C p(l)} \quad (2)$$

where $p_\lambda(\mathbf{s}|l(\mathbf{s}))$ is the likelihood of the score vector $\mathbf{s}$ given the true target language $l(\mathbf{s})$ and model parameters λ, $p(l)$ is the probability of language l and C is an heuristic factor. In this work, C has been set to 10 and 20 MMI iterations have been conducted.

- **Fully-Bayesian Gaussian Backend (FBGB)**: Under the generative, fully Bayesian Gaussian backend paradigm [5], the distribution of trained languages scores is modeled by a multivariate normal distribution $N(\mu_t, \Sigma)$ too, but instead of using a maximum likelihood estimate of the model parameters, it integrates over all possible parameters (according to their respective priors).
- **Generative Gaussian Mixture Backend (GMB)**: The Gaussian backend model is augmented to a Gaussian mixture, while a ML full covariance matrix Σ is shared across all target languages. Mixture means are estimated using the Expectation-Maximization (EM) algorithm.
- **Logistic Regression Backend (LR)**: Regularized Multiclass Logistic Regression (MLR) can be used to train an affine transform:

$$\hat{\mathbf{s}} = \mathbf{C}\mathbf{s} + \mathbf{d} \quad (3)$$

  Note that equations (1) and (3) are basically the same, the only difference being the estimation criteria of the affine transform.
- **Z-norm and T-norm score normalization**: Score normalization techniques such as Z-norm and T-norm [6], can help reducing the environmental effects on the score space. Nevertheless, they are rarely used alone in Language Recognition systems, but instead they are usually applied before some other backend. Impostor development signals are used to estimate language dependent means and variances of the Z-norm. T-norm means and variances are estimated for each test signal and language score, using the rest of the scores (of the same test signal) as impostors.

Backend output score vectors are further calibrated and fused according to a discriminative linear model which minimizes the so called C_{LLR} function (more precisely C_{mxe}, the multiclass cross entropy) on the development set, by means of logistic regression under a multiclass paradigm [7]. After the fusion, well-calibrated log-likelihoods are obtained, for which a minimum expected cost Bayes decision threshold is applied according to application-dependent language priors and costs.

Backend and fusion parameters have been separately estimated on the development set, and then applied to the corresponding segments in the evaluation set. The *FoCal* toolkit has been used to estimate and apply the fusion models [8].

3. Experimental Setup

3.1. Training, development and evaluation datasets

Training and development data were limited to those distributed by NIST to all 2009 LRE participants [9]: (1) conversational telephone speech from previous LREs: the Call-Friend Corpus, the OHSU Corpus provided by NIST for LRE05 and the development corpus provided by NIST for the 2007 LRE; and (2) narrow band (telephone channel) speech segments from *Voice Of America* (VOA) broadcast news recordings (provided by NIST for the 2009 LRE).

A set of 66 languages/dialects was defined. Each of them was mapped either to a target language of the NIST 2009 LRE or to Out-Of-Set (OOS). For example, Mainland and Taiwan from the NIST 2007 LRE and Mandarin from VOA were all mapped to Mandarin, whereas Arabic was mapped to OOS. Persian and Farsi were mapped to the same language, as was properly pointed in [10]. For the languages appearing in VOA recordings, the longest speech segment out of each file was posted to the training dataset, with a minimum of 225 segments per language. The number of segments extracted per file was relaxed (augmented) for those languages with few files in VOA. The training dataset consisted of 43278 segments, which amounted to 2286 hours. For development, speech segments lasting around 30 seconds (between 25 and 35 seconds) were randomly extracted, using no more than 2 segments per file, and a minimum of 225 segments per language. Segments of 30-seconds of the evaluation set of NIST 2007 LRE were also considered. The development dataset consisted of 13269 segments, which amounted to around 380 hours. Evaluation was carried out on the NIST 2009 LRE evaluation corpus, specifically on the 30-second, closed-set condition (primary evaluation task).

3.2. Language Recognition Systems

3.2.1. Acoustic Subsystems

Acoustic features consisted of the concatenation of 7 Mel-Frequency Cepstral Coefficients and the Shifted Delta Cepstrum coefficients [11] under a 7-2-3-7 configuration. A gender independent 1024-mixture Gaussian Mixture Model (GMM) was used as Universal Background Model (UBM) and zero-order and centered and normalized first-order Baum-Welch statistics were computed for each input utterance.

The first acoustic subsystem followed the Linearized Eigenchannel GMM (LE-GMM) approach (also known as *Dot-Scoring*), which makes use of a linearized, channel compensated and normalized approximation of the likelihood ratio in the GMM-UBM approach to score test segments against target models [12]. The second acoustic subsystem followed the Total Variability generative iVector approach, as described in [13].

3.2.2. Phonotactic Subsystems

Three phonotactic sub-systems were developed under a phone-lattice Support Vector Machine (SVM) approach. Given an input signal, an energy-based voice activity detector was applied in first place, which split and removed long-duration non-speech segments. Then, the Temporal Patterns Neural Network (TRAPs/NN) phone decoders developed by the Brno University of Technology (BUT) for Czech, Hungarian and Russian [14], were applied to perform phone tokenization. Regarding channel compensation, noise reduction, etc. the three sub-systems relied on the acoustic front-end provided by BUT decoders.

BUT decoders were configured to produce phone posteriors that were converted to phone lattices by means of HTK [15] along with the BUT recipe, on which expected counts of phone n-grams were computed using the *lattice-tool* of SRILM [16]. Finally, a SVM classifier was applied, SVM vectors consisting of counts of features representing the phonotactics of an input utterance. In this work, phone n-grams up to $n = 4$ were used, weighted as in [17]. L2-regularized L1-loss support vector classification was applied, by means of LIBLINEAR [18], whose source code was slightly modified to get regression values.

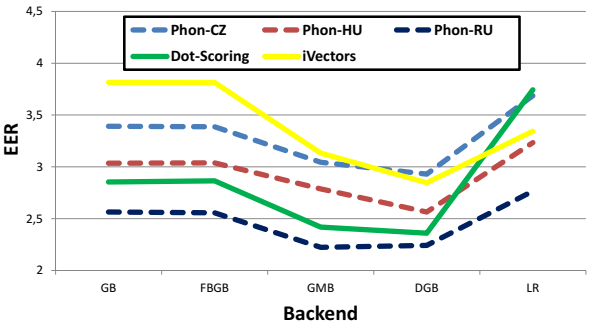

Figure 1: EER for the five state-of-the-art subsystems using different backends.

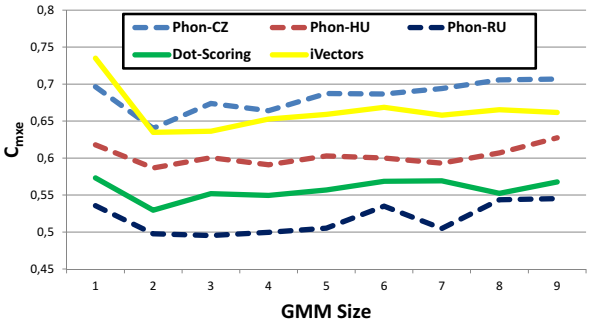

Figure 2: C_{mxe} for the five subsystems using a generative Gaussian mixture backend, the number of the components of the mixture ranging from 1 to 9.

3.3. Evaluation measures

In this work, systems will be compared in terms of Equal Error Rate (EER), one of the most common ways of comparing the performance of language recognition systems, but also in terms of the so called C_{LLR} (more precisely C_{mxe}, the multiclass cross entropy) [7], an alternative performance measure used in NIST evaluations.

4. Results

Figure 1 shows EER results for the five subsystems using different backends. For many of the backends, the relative performance remains constant, being the Russian phonotactic subsystem (Phon-RU) the *winner*, followed by the Dot-Scoring subsystem. Regarding the backends, the fully Bayesian Gaussian backend (FBGB) shows an almost imperceptible improvement with respect to the generative Gaussian backend (GB). The Bayesian estimation is supposed to guard against overfitting, which could indicate that the development set is large enough to estimate a simple GB. The generative Gaussian mixture backend (GMB) improves the performance in all the cases, achieving the best results with Phon-RU. The discriminative Gaussian backend (DGB) gets the best overall performance, but at the expense of having to tune the MMI factor and the number of training iterations, and at the risk of overfitting. Last, the multiclass Logistic Regression backend (LR) didn't get the expected results, maybe due to the lack of an optimum estimation algorithm.

The GMB backend in Figure 1 is composed by two components per mixture. Figure 2 shows how the performance degrades as the number of components increases, being 2 the most conservative size.

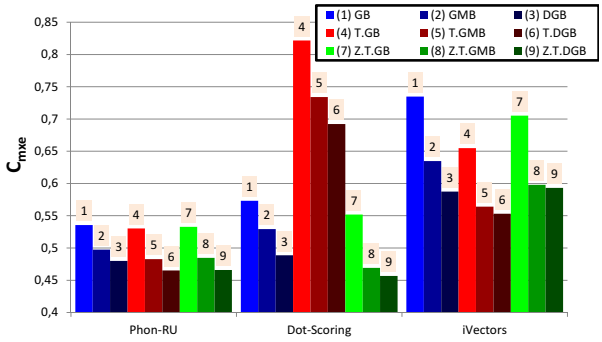

Figure 3: C_{mxe} for the best phonotactic and two acoustic subsystems using *T-norm* and *ZT-norm* prior to different backends.

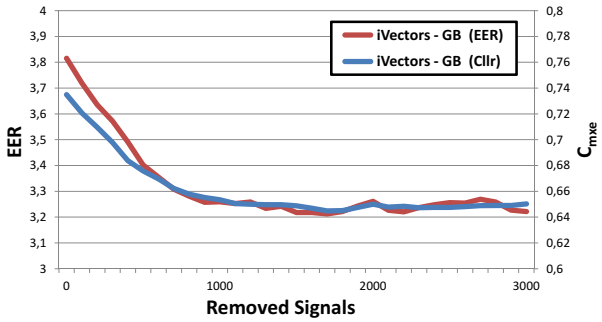

Figure 4: EER and C_{mxe} for the i-Vector subsystem using a generative Gaussian backend, when the highest scored development signals are discarted.

Figure 3 shows the effect of applying Z-norm and T-norm for the three best performing backends and subsystems. Note that a Z-norm has no effect if it precedes any Gaussian model, so only the T-norm and the ZT-norm are considered. The Phon-RU subsystem shows a very slight improvement when either a T-norm or the ZT-norm are applied. On the other hand, and despite being quite similar approaches, the behavior of the two acoustic systems differs when the score normalizations are applied. The Dot-Scoring subsystem degrades when the T-norm is applied, while it improves with the ZT-norm. The iVector subsystem, however, improves with the T-norm and shows an irregular behavior under the ZT-norm.

4.1. Filtering out development signals

A well designed development set should not contain speakers that were previously used to train the language models. Nevertheless, when uncontrolled recordings (such as VOA broadcast news recordings) are used, this requirement may be unenforceable. Repeated speakers should have high likelihoods, and therefore, a straightforward cleaning method could be to remove the highest score signals from the development set. Figure 4 shows that removing such signals clearly improves the iVector subsystem's performance. Since an excessive reduction could affect the robustness of the more complex backends, removing the 1200 highest likelihood signals (out of 13269) seems to be appropriate. Figure 5 shows the histogram of the fused development target scores, which clearly features a stretched tail to the right. The 1200 highest scores are just those above 16.5 (marked in red), a threshold that fits quite well the right tail.

Table 1 summarizes EER results for each subsystem and the fusion of all of them, aplying the three best performing backends, with and without score normalization and filtering. Best

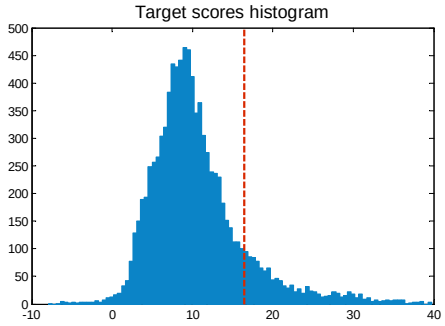

Figure 5: Histogram of fused development target scores. The 1200 highest scores are above 16.5 (marked in red).

Table 1: EER for each subsystem and fusion aplying the three best performing backends, with and without score normalization and filtering.

				Norm & Filtering		
	GB	GMB	DGB	GB	GMB	DGB
Phon-CZ	3.39	3.05	2.93	3.34	2.92	2.88
Phon-HU	3.04	2.79	2.56	2.85	2.71	2.43
Phon-RU	2.56	2.22	2.24	2.48	2.34	2.19
Dot-Scoring	2.85	2.42	2.36	2.63	2.15	2.13
iVectors	3.82	3.13	2.85	2.88	2.68	2.56
Fused	1.62	1.41	1.39	1.47	1.25	1.32

performance was attained by GMB (1.25 EER) when the score normalization and filtering was used, yielding 23% relative improvement with respect to GB (1.62 EER) when no score normalization or filtering was used.

5. Conclusions

A study of different backends has been carried out in a state-of-the-art Language Recognition system. The backend plays a dual role, mapping the trained language scores to the target languages and adapting and calibrating the score space. The 2-component Gaussian Mixture Backend and the Discriminative Gaussian Backend attained the best performance, being the GMB a more conservative alternative. Both T-norm and ZT-norm may improve results, but their effect was shown to be different when applied to quite similar subsystems. Filtering out high score development signals (probably avoiding repeated speakers) was in practice a simple and efficient method to improve performance.

6. Acknowledgments

This work has been supported by the University of the Basque Country, under grant GIU10/18 and project US11/06, by the Government of the Basque Country, under program SAIOTEK (project S-PE11UN065), and the Spanish MICINN, under *Plan Nacional de I+D+i* (project TIN2009-07446, partially financed by FEDER funds). Mireia Diez is supported by the Department of Education, Universities and Research of the Government of the Basque Country, under a 4-year research fellowship.

7. References

[1] E. Singer, P. A. Torres-Carrasquillo, T. P. Gleason, W. M. Campbell, and D. A. Reynolds, "Acoustic, phonetic and discriminative approaches to automatic language identification," in *Proceedings of Eurospeech (Interspeech)*, Geneva, Switzerland, 2003, pp. 1345–1348.

[2] M. F. BenZeghiba, J. L. Gauvain, and L. Lamel, "Language score calibration using adapted Gaussian back-end," in *Proceedings of Interspeech 2009*, Brighton, UK, September 2009, pp. 2191–2194.

[3] N. Dehak, A. McCree, D. Reynolds, F. Richardson, E. Singer, D. Sturim, and P. Torres-Carrasquillo, "MITLL 2011 Language Recognition Evaluation System Description," in *Proceedings of the NIST 2011 Language Recognition Evaluation Workshop*, Atlanta (GA), USA, December 2011.

[4] N. Brümmer, S. Cumani, O. Glembek, Karafiat, P. Matejka, J. Pesan, O. Plchot, M. Soufifar1, and E. de Villiers, "Brno276 System Description for NIST LRE 2011," in *Proceedings of the NIST 2011 Language Recognition Evaluation Workshop*, Atlanta (GA), USA, December 2011.

[5] N. Brümmer, "Generative, Fully Bayesian, Gaussian Pattern Classifier," AGNITIO LABS, South Africa, Tech. Rep., September 2011. [Online]. Available: https://sites.google.com/site/nikobrummer/

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

[7] N. Brümmer and D. van Leeuwen, "On calibration of language recognition scores," in *Proceedings of Odyssey - The Speaker and Language Recognition Workshop*, 2006, pp. 1–8.

[8] FoCal, *Toolkit for Evaluation, Fusion and Calibration of statistical pattern recognizers*, 2008, http://sites.google.com/site/nikobrummer/focal.

[9] A.Martin and C. Greenberg, "The 2009 NIST Language Recognition Evaluation," in *Proceedings of Odyssey 2010: The Speaker and Language Recognition Workshop*, Brno, Czech Republic, 2010, pp. 165–171.

[10] Z. Jancik, O. Plchot, N. Brummer, L. Burget, O. Glembek, V. Hubeika, M. Karafiat, P. Matejka, T. Mikolov, A. Strasheim, and J. Cernocky, "Data Selection and Calibration Issues in Automatic Language Recognition - Investigation with BUT-AGNITIO NIST LRE 2009 System," in *Proc. Odyssey 2010 - The Speaker and Language Recognition Workshop*, 2010, pp. 215–221.

[11] P. A. Torres-Carrasquillo, E. Singer, M. A. Kohler, R. J. Greene, D. A. Reynolds, and J. R. Deller, "Approaches to language identification using Gaussian mixture models and Shifted Delta Cepstral features," in *Proceedings of ICSLP*, 2002, pp. 89–92.

[12] N. Brümmer, A. Strasheim, V. Hubeika, P. Matejka, L. Burget, and O. Glembek, "Discriminative Acoustic Language Recognition via Channel-Compensated GMM Statistics," in *Proceedings of Interspeech*, Brighton, UK, September 2009, pp. 2187–2190.

[13] D. Martínez, O. Plchot, L. Burget, O. Glembek, and P. Matejka, "Language Recognition in iVectors Space," in *Proceedings of the Interspeech*, Firenze, Italy, 2011, pp. 861–864.

[14] P. Schwarz, "Phoneme recognition based on long temporal context," Ph.D. dissertation, Faculty of Information Technology BUT, http://www.fit.vutbr.cz, Brno, CZ, 2008.

[15] S. Young, G. Evermann, M. Gales, T. Hain, D. Kershaw, X. Lui, G. Moore, J. Odell, D. Ollason, D. Povey, V. Valtchev, and P. Woodland, *The HTK Book (for HTK Version 3.4)*, Cambridge, UK, 2006.

[16] A. Stolcke, "SRILM - an extensible language modeling toolkit," in *Proceedings of ICSLP*, November 2002, pp. 257–286.

[17] F. Richardson and W. Campbell, "Language recognition with discriminative keyword selection," in *Proceedings of ICASSP*, 2008, pp. 4145–4148.

[18] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin, "LIBLINEAR: A library for large linear classification," *Journal of Machine Learning Research*, vol. 9, pp. 1871–1874, 2008, software available at http://www.csie.ntu.edu.tw/∼cjlin/liblinear.