INTERSPEECH
2009 BRIGHTON

Language Score Calibration using Adapted Gaussian Back-end

Mohamed Faouzi BenZeghiba, Jean-Luc Gauvain and Lori Lamel

Spoken Language Processing Group
LIMSI - CNRS B.P. 133 91403 ORSAY CEDEX FRANCE

Abstract

Generative Gaussian back-end and discriminative logistic regression are the most used approaches for language score fusion and calibration. Combination of these two approaches can significantly improve the performance. This paper proposes the use of an adapted Gaussian back-end, where the mean of the language-dependent Gaussian is adapted from the mean of a language-specific background Gaussian via *maximum a posteriori* estimation algorithm. Experiments are conducted using the LRE-07 evaluation data. Compared to the conventional Gaussian back-end approach for a *closed set* task, relative improvements in the C_{avg} of 50%, 17% and 4.2% are obtained on the $30s$, $10s$ and $3s$ conditions, respectively. Besides this, the estimated scores are better calibrated. A combination with logistic regression results in a system with the best calibrated scores.

Index Terms: Language recognition, Gaussian back-end, Adaptation

1. Introduction

Language detection is a binary decision of whether the language of a speech segment corresponds to a specific language from a set of target languages. In any decision making task, producing the correct decision is essential, but reporting the confidence with which the decision is made is also important. The confidence measure provides information about how reliable the decision is. In real applications, useful systems need not only be accurate in terms of classification, but also they need to be well calibrated. When two systems have the same classification performance, the more calibrated one should be used.

State-of-the-art language recognizers typically make use of several acoustic and phonotactic sub-systems. Combining the outputs of these sub-systems, generally improves the performance. The combined system is more accurate and the estimated scores are better calibrated. Score calibration consists of mapping the original scores to a new ones that are reliable estimates of the true class probabilitites. Recently, several score fusion and calibration techniques have been proposed for language recognition task, including Gaussian back-end [1] [2], logistic regression [3] [4], combination of these two techniques [5] [6] [7], neural network [8] and support vectors machine [9]. In this latter reference, a comparison study between most of these techniques can be found. A toolkit known as *the FoCal Multi-class toolkit*[1] that implements the first two approaches is also available. More details about the topic of score calibration for language recognition can be found in [4].

In previous work [10], a technique for *open-set* language detection with Gaussian back-end was proposed. The mean of the target-dependent Gaussian was adapted (using *maximum a posteriori* adaptation) from the mean of a common Gaussian Background model trained using data from target languages only and used to represent the out-of-set languages. Competitive results to the state-of-the art approaches were obtained, although, the assumption was made that no data from out-of-set (OOS) languages is available. This paper further investigates this approach for the *closed set* task.

2. Language score fusion and calibration

Figure 1 shows a block diagram of the score fusion and calibration module. Experiments are conducted using the FoCal Multi-class toolkit.

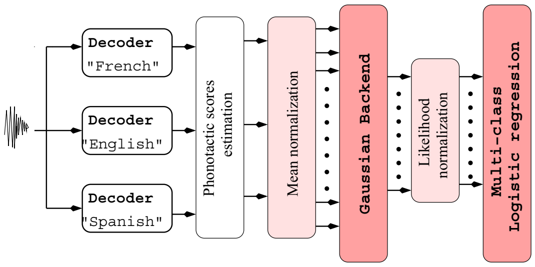

Figure 1: *A block diagram of the score fusion and calibration module*

The language recognition system makes use of the parallel Phone Recognizer followed by Language Modeling (PPRLM) approach [11]. Phonotactic scores estimated using each decoder are first mean normalized. This step is done for each decoder independently. The normalized scores are then stacked in a feature vector of dimension $d = N_D * N_{lm}$ (number of decoders times number of phonotactic language models).

2.1. Gaussian back-end approach

The set of feature vectors associated with a given target language are used to train a language dependent multivariate normal distribution $N(\mu_\ell, \Sigma_\ell)$ (one Gaussian). In this work, all Gaussians share a common full covariance matrix and form what is called the *Gaussian Back-end* (GB). The decision function can be seen as an affine transform [3] expressed as follows [10]:

$$\delta_\ell(\mathbf{x}) = (\mathbf{\Sigma}^{-1}\mu_\ell)^{\mathbf{t}}\mathbf{x} - \frac{\mathbf{1}}{\mathbf{2}}\mu_\ell^{\mathbf{t}}\mathbf{\Sigma}^{-1}\mu_\ell \qquad (1)$$

where μ_ℓ is the mean vector, and Σ is the common covariance matrix. Because of this linearity, this technique is known as

*This work was in part supported by OSEO under the Quaero program.

[1] http://niko.brummer.googlepages.com/focalmulticlass

Linear Gaussian back-end. It transforms d general scores to N_l multi-class log-likelihoods (N_l is the number of target languages). No explicit LDA is needed since this transformation implicitly performs it. In this case, Gaussian back-end performs both score fusion and calibration.

2.2. Adapted Gaussian Back-end approach

In this work, only the adaptation of the mean will be considered. The adaptation of the covariance matrix was found to be not effective, although, the decision function is no longer linear. The general form of the adaptation can be expressed as follows [13]:

$$\hat{\mu}_\ell = \alpha_\ell \mu_\ell + (1 - \alpha_\ell)\bar{\mu} \qquad (2)$$

where μ_ℓ and $\hat{\mu}_\ell$ are the mean of language dependent Gaussian before and after adaptation, and $\bar{\mu}$ is the mean of the Gaussian background model. The effectiveness of this adaptation depends on several factors, such as the representation of the background model, the choice of the adaptation factor and the optimization procedure.

The adaptation factor α_ℓ is usually chosen to be common for all classes (languages). This is a sub-optimal selection, as the available development data is not well balanced between different languages. For example, in our development data, the number of speech segment with $30s$ is equal to 2485 for English and only 43 for Thai. In this work, the factor α_ℓ is defined as follows:

$$\alpha_\ell = \frac{n_\ell}{n_\ell + \rho} \qquad (3)$$

where n_ℓ is the number of examples for the target language ℓ and ρ is the relevant factor to be optimized. This definition makes α_ℓ language dependent and more finely optimized.

In previous work, a common Gaussian trained using data from all target languages was used as background model for adaptation. In this work, a language-specific background model was used. The mean of this model is estimated using data from the non-target languages only. To avoid the mean to be biased to languages with big amount of development data, the mean $\bar{\mu_\ell}$ of the target-specific background model is estimated as follows:

$$\bar{\mu_\ell} = \frac{1}{N_l - 1} \sum_{L_T,\, q \neq \ell} \mu_q \qquad (4)$$

where L_T is the set of target languages. Using this definition, performances was consistently better for all test conditions.

The optimization of the factor ρ was performed using a stratified k-fold cross-validation with k equal 5, to make sure that each fold contains the same proportion of class labels as in the original data. Therefore, for each class (target language) there was 20% of the original data in each fold. The selected value of the parameter ρ is the one that minimizes the average cross-entropy referred to as *multi-class* C_{llr} [4]. For a given fold k, the multi-class C_{llr} measure is defined as follows:

$$C^k_{llr} = -\frac{1}{N_s \log_2} \sum_{L_T} \frac{1}{n_{\ell_k}} \sum_{s=1}^{n_{\ell_k}} \log_2 P_s \qquad (5)$$

where N_s is the total number of test segment, n_{ℓ_k} is the number of test segments for the target language ℓ in the fold k and P_s is the *posterior probability of the true class of trial* s defined using softmax as follows:

$$P_s = \frac{\exp(p(s|c(s))}{\sum_{L_T} \exp(p(s|\ell))} \qquad (6)$$

where $p(s|\ell)$ is the *likelihood* of the trial s given the language ℓ, and $c(s)$ is the true class (language) of trial s. It is worth mentioning here that minimizing the C_{llr} helps estimating better calibrated scores, but it does not necessary make the classifier more accurate[2]

2.3. Combination with multi-class logistic regression

If the amount of development data is big enough, language *log likelihoods* at the outputs of the Gaussian back-end can be further calibrated using a discriminative multi-class Logistic Regression (MLR) [5] [6] [7]. As implemented in the FoCal toolkit, the calibration transformation includes one scale parameter (a positive scalar β) and N_L-dimension translation vector $\vec{\gamma}$. These parameters are optimized according to the *multi-class* C_{llr} (5). The final language *log likelihood* is estimated as follows:

$$\log \hat{p}(s|\ell) = \beta \log p(s|\ell) + \gamma_\ell \qquad (7)$$

where $\log p(s|\ell)$ is the language *log likelihood* estimated by the language-dependent Gaussian.

In case of conventional Gaussian back-end, language *log likelihoods* are first converted to a *log likelihood ratio* (LLR) by normalizing each language likelihood with respect to the other likelihoods. This normalization leads to some improvements, in particular when phonotactic scores are mean normalized. However, we found that this normalization is not effective and might degrade the performance with the adapted Gaussian back-end. In this case, language *log likelihoods* are used as they are as inputs to the logistic regression. Results reported in this work are based on these findings.

3. Experimental set-up

3.1. Data description and pre-processing

Table 1 specifies the databases from which training data (used to generate phonotactic language models) and development data (used to train the fusion and calibration module) are selected. These data sets were defined by MIT Lincoln Labs when developing their NIST LRE 2007 system [5].

TRAIN DATA FOR LM	DEV.DATA FOR FUSION MODULE	EVAL. DATA
LRE-96 train+dev	lid96e1,lid03e1	
NIST LRE07 train	lid05e1, lid07d1	*lid07e1*
Callhome, Mixer	Callhome	14 *languages*
Fisher	Fisher, Mixer	

Table 1: *Databases used to select training and development data. Evaluation performed on the NIST LRE-07 evaluation data set (lid07e1).*

Performaces are evaluated using the NIST LRE-07[3] evaluation data sets. The task of interest is the *closed set* language detection. There are 14 target languages, and about 2155 test segments for each duration conditions. Speech segments were mainly extracted from the Fisher, Mixer, Callfriend and OGI corpora.

Standard 12 PLP coefficients with energy are extracted every 10 ms, with a 30 ms window. Cepstral mean removal and

[2] In general, a better calibrated classifier is a more accurate classifier.

[3] http://www.nist.gov/speech/tests/lang/2007/

variance normalization are applied to each segment. These features are augmented by their first and second derivatives, resulting in a 39 dimensional feature vector. Speech activity detection was carried out using Gaussian mixture models to segment the audio signal into speech/non-speech regions. Two Gaussian mixtures, one for speech and one for non-speech with 2048 and 512 mixtures, respectively, were used.

3.2. System description

The PPRLM system uses 3 context-dependent phone decoders for English, French and Spanish. Each model covers about 3000 phone contexts, with 3000 tied states and a mixture of 32 Gaussians per state. Constrained MLLR adaptation was performed to improve phone lattice decoding. Back-off *4*-gram phonotactic models are generated from phone lattices with Witten-Bell discounting using the SRILM toolkit.[4] Multiple phonotactic models per decoder are generated for languages with several data sources [12]. More details about this system and its performance can be found in [7, 10].

3.3. Detection score estimation

The detection decision is made based on the *detection log likelihood ratio* (llr) defined as follows:

$$llr(s|\ell) = \log \left[\frac{P_{tar} . p(s|\ell)}{\sum_{L_T,\, q \neq \ell} P_{non-tar} . p(s|q)} \right] \quad (8)$$

where $p(s|\ell)$ is the likelihood of the test trial s given the language ℓ. It can be the outputs of the Gaussian backend or the multi-class logistic regression (MLR). The target language *prior* P_{tar} is equal to 0.5. The $P_{non-tar}$ is equal to:

$$P_{non-tar} = (1 - P_{tar})/(L - 1) \quad (9)$$

The $llrs$ are then compared to the theoretical threshold $\Delta = 0$ to make a decision. Results are reported in terms of C_{avg} as defined by NIST[5] and the *multi-class* C_{llr} as defined in (5).

4. Experimental results and discussion

4.1. Using Gaussian back-end only

In our system, there are 26 phonotactic models and 3 decoders, therefore, the dimension of the feature vector is equal to 78. Table 2 reports the results in terms of C_{avg} and *multi-class* C_{llr} on the $30s$, $10s$ and $3s$ conditions for the two Gaussian back-end approaches.

Results show that the proposed approach outperforms the conventional approach, in particular for long test segments. In terms of C_{avg}, the relative improvement on the $30s$, $10s$ and $3s$ conditions, is 50%, 17% and 4.2%, respectively. For the $30s$ segments this is a considerable gain. The same trend can be observed for the C_{llr} measure, which means that with adaptation not only detection results are improved but the detection scores are better calibrated. In examining the false acceptance and false rejection errors produced by the two approaches, both kinds of errors are reduced except for the $3s$ condition where there is a small increase in false acceptances. More importantly the false rejections are reduced more than the false acceptances.

This improvement can be explained as follows: The adaptation by (2) consists of shifting the target class mean towards the non-target class mean (here grouped and represented by one Gaussian). The amount of this shift is determined by the parameter α_ℓ. In the feature space (score vectors), this results in an increase in the region shared between the two classes. Increasing the confusion region is critical and can degrade the performance if the class parameters are badly estimated. Because this is not usually the case, it turns out to be beneficial. This adaptation can also be seen as adding some information about the non-target language characteristics into the target language Gaussian.

Dur.	GAUSSIAN BACK-END APPROACH	$C_{avg[\%]}$	C_{llr}
$30s$	CONVENTIONAL	2.7	0.553
	ADAPTED ($\rho = 19.5$)	**1.3**	**0.245**
$10s$	CONVENTIONAL	7.0	1.030
	ADAPTED ($\rho = 17.5$)	**5.8**	**0.839**
$3s$	CONVENTIONAL	16.8	2.092
	ADAPTED ($\rho = 30$)	**16.2**	**1.975**

Table 2: *Performances of the conventional and adapted Gaussian back-end on the lid07e1 data set and for different duration conditions.*

Indeed, we have observed that when target segments are correctly detected (classified), most of the time, the target score is relatively high compared to the best non-target score. (i,e; both feature vectors are far from the decision surface). In this case, the language likelihood will be slightly modified without affecting the original decision. However when a target segment is missed and accepted as another language (non-target), the difference in the two scores is rather small. Therefore, perturbing slightly target and non-target scores can lead to a change in the decision.

Comparing different background model representations and their means estimation, we found that when a common background model is used, or the mean of language-specific background models are estimated according to the statistical formula, the C_{avg} on the $30s$, $10s$ and $3s$ conditions is equal to 1.4%, 5.9% and 16.3%, respectively. However, when a common adaptation factor α is used, the C_{avg} is equal to 1.5%, 5.8% and 16.5% on the above conditions, respectively. Although, differences are not significant, the proposed choices systematically give better results.

4.2. Combination with logistic regression

It should be recalled that in case of the conventional GB, language log likelihoods are first converted to log likelihood ratios which are used as inputs to the logistic regression, while this conversion found to harm performance in case of the adapted GB. Table 3 reports the results of the two Gaussian back-end approaches combined with multi-class logistic regression. The value of the scale parameter in (7) is also reported. The following observations can be drawn:

First, further score calibration with logistic regression is more beneficial to the conventional GB approach than with the adapted GB approach. The scale factor β is always lower with the conventional GB approach than with the adapted GB, indicating that language *log likelihoods* estimated with the conventional GB are somewhat over-confident [4]. Second, in comparing the results of the two combined approaches to those obtained with the adapted GB only (Table 2), we observe that there

[4] http://www.speech.sri.com/projects/srilm/

[5] http://www.nist.gov/speech/tests/lang/2007/LRE07EvalPlan-v8b.pdf

DUR.	GB+MLR APPROACH	C_{avg}	C_{llr}
$30s$	CONVENTIONAL ($\beta = 0.4$) ADAPTED ($\beta = 0.6$)	1.3 1.3	0.206 **0.199**
$10s$	CONVENTIONAL ($\beta = 0.5$) ADAPTED ($\beta = 0.8$)	5.5 5.5	0.759 **0.754**
$3s$	CONVENTIONAL ($\beta = 0.6$) ADAPTED ($\beta = 0.9$)	16.3 **16.2**	**1.932** 1.940

Table 3: *Performance of the conventional and adapted Gaussian backend combined with multi-class logistic regression.*

is no significant differences between the two approaches, although scores estimated by the combined approaches are better calibrated. Third, the language scores estimated by the combined adapted GB+MLR approach are at least as well calibrated as the combined conventional GB+MLR approach. As a result, the performance in terms of C_{avg} obtained with the former approach is equivalent or better than the latter one.

4.3. Effect of the size of dev data

The performance of the fusion and calibration modules improve as the amount of dev data increases. For the adapted GB approach, the dev data is used for training the background model and for adaptation. Combination with MLR improves the performance if the amount of dev data is large enough. To study the effect of the amount of dev data on the C_{avg} measure, four data sets are created from the original dev data. Figure 2 plots the variations of C_{avg} as a function of the amount of dev data on the $30s$ condition.

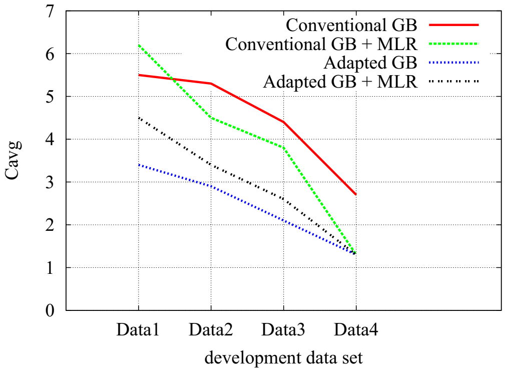

Figure 2: *Variations of C_{avg} as a function of the amount of dev data for the* $30s$ *condition, Data1=lid96e1+lid07d1, Data2 = Data1+lid03e1, Data3 = Data2+lid05e1, Data4 = all dev data*

It can be observed that when the amount of dev data is small, the adapted GB approach alone outperforms significantly the other approaches. As the amount of dev data increases, the difference in C_{avg} gets reduced, but the adapted GB still performs as well as the best approach. If only previous evaluation data sets (Data3) provided by NIST are used for development, then the adapted GB alone performs the best.

5. Conclusion

This paper proposed and analysed the use of the adapted Gaussian back-end for language score fusion and calibration. Compared to the conventional Gaussian back-end, significant improvements in the C_{avg} measure were obtained for a *closed-set* task and for all duration conditions (up to 50% relative on the $30s$ condition). Similar trends were observed in the *multi-class* C_{llr} measure, indicating that decision scores were also better calibrated. Combination with multi-class logistic regression was also investigated. This combination found to be more beneficial for the conventional back-end approach. With the proposed approach, the combination gave performances at least equivalent to the previous approach with both C_{avg} and C_{llr} measures. The proposed adapted Gaussian back-end is found to be more effective than the traditional methods when the amount of development data is small, which is usually the case.

6. Acknowledgments

We would like to thank MIT Lincoln Labs for sharing their training and development data lists with us.

7. References

[1] M. A. Zissman "Predicting , Diagnosing and Improving Automatic Language Identification Performance", *Eurospeech'97*, Volume 1 pages 51 - 54

[2] E. Singer, et al. "Acoustic, Phonetic, and Discriminative Approaches to Automatic Language Identification", *Interspeech'03:* 1345-1348.

[3] D.A. van Leeuwen and N. Brummer, "Channel-dependent GMM and Multi-class Logistic Regression models for language Recognition" *2006 IEEE Odyssey: The Speaker and Language Recognition Workshop*.

[4] N. Brummer and D.A. van Leeuwen, "On Calibration of language recognition scores" *2006 IEEE Odyssey: The Speaker and Language Recognition Workshop*, pp. 1-8.

[5] P. A. Torres-Carrasquillo et al. "The MITLL NIST LRE 2007 Language Recognition system" *Interspeech'08* pp. 719-722.

[6] P. Matejka et al. "BUT Language Recognition System for NIST 2007 Evaluations" *Interspeech'08*, pp. 739-742.

[7] M.F. BenZeghiba, J.L. Gauvain and L. Lamel, "Context-Dependent Phone models and Models Adaptation for Phonotactic Language Recognition", *Interspeech'08*, pp. 313-316.

[8] J. L. Gauvain, A. Messaoudi and H. Schwenk, "Language Recognition Using Phone Lattices", *ICSLP'04*

[9] H. Suo, M. Li, P. Lu and Y. Yan, "Using SVM as Back-end Classifier for Language Identification" *EURASIP Journal on Audio, Speech and Music Processing*, Vol. 2008

[10] M.F. BenZeghiba, J.L. Gauvain and L. Lamel, "Gaussian Back-end Design for Open-set Language Detection" *ICASSP'09*, pp. 4349-4352.

[11] M. A. Zissman, "Comparison of Four Approaches to Automatic Language Identification of Telephone Speech", IEEE Trans. Speech and Audio Proc., 4(1):31-44, 1996.

[12] O. Glembek, P. Matejka, L. Burget and T. Mikolov "Advances in Phonotactic Language Recognition" *Interspeech'08*, pp. 743-746.

[13] J. L Gauvain and C. H. Lee "Maximum a-posteriori estimation for multivariate Gaussian mixture observations of Markov chain" IEEE Trans. Speech and Audio Proc., 2:31-44, 1994.