INTERSPEECH 2017
August 20–24, 2017, Stockholm, Sweden

Calibration Approaches for Language Detection

Mitchell McLaren[1], Luciana Ferrer[2], Diego Castan[1], Aaron Lawson[1]

[1]Speech Technology and Research (STAR) Laboratory, SRI International, Menlo Park, USA
[2]Instituto de Investigación en Ciencias de la Computación (ICC), CONICET-UBA, Argentina.
{mitch, aaron, dcastan}@speech.sri.com, lferrer@dc.uba.ar

Abstract

To date, automatic spoken language detection research has largely been based on a closed-set paradigm, in which the languages to be detected are known prior to system application. In actual practice, such systems may face previously unseen languages (out-of-set (OOS) languages) which should be rejected, a common problem that has received limited attention from the research community. In this paper, we focus on situations in which either (1) the system-modeled languages are not observed during use or (2) the test data contains OOS languages that are unseen during modeling or calibration. In these situations, the common multi-class objective function for calibration of language-detection scores is problematic. We describe how the assumptions of multi-class calibration are not always fulfilled in a practical sense and explore applying global and language-dependent binary objective functions to relax system constraints. We contrast the benefits and sensitivities of the calibration approaches on practical scenarios by presenting results using both LRE09 data and 14 languages from the BABEL dataset. We show that the global binary approach is less sensitive to the characteristics of the training data and that OOS modeling with individual detectors is the best option when OOS test languages are not known to the system.

1. Introduction

In recent years, language recognition technology has received increased attention, with a focus on real-world situations, such as short durations, high noise and efficient processing [1]. Benchmarks for language recognition technology have largely focused on the case of *closed-set* language recognition evaluations, in which the languages that will be encountered by the recognition system are known prior to testing and only those languages are modeled. This closed-set paradigm enables the system to leverage information from each language detector to make the best decision for a given test sample [2]. In contrast to closed-set language recognition, this paper focuses on *open-set* detection, in which many of the test languages are unseen yet still must be rejected by the system. Coping with out-of-set (OOS) test languages often involves training a language detector on data pooled from all languages (target and non-target) available during training [2]. This approach, however, does not account for some of the practical limitations of language recognition including the desire to detect as many languages as possible which hinders the ability to create an explicit OOS model.

From a technology point of view, most recent publications on language recognition, such as [3, 4, 5, 6, 7] have focused on a similar system framework as depicted in Figure 1 and detailed in the following section. This framework jointly models languages such that information is shared across the models through a multi-class objective function. In a practical context, the shortcomings of this approach include (1) the assumption that languages for all spoken languages detectors (modeled languages) are observed during use with known priors, and (2) using the multi-class objective to optimize the calibration of the likelihoods rather than directly targeting the calibration of the final scores that are ultimately used to make decisions on language content for the detection task.

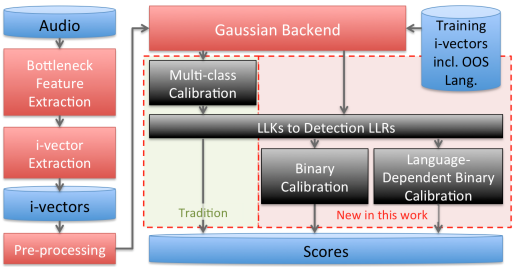

Figure 1: *The framework considered in this work. The binary objective function aims to directly target the calibration of the final scores for the practical application of language detection.*

In this paper, we focus on applying calibration with a binary objective function [8] after converting the scores into detection LLRs such that we directly calibrate the scores that are used for language detection. Additionally, binary calibration enables for more relaxed system constraints. We also demonstrate the limitations of OOS modeling when the OOS test languages are unknown to the system in all considered calibration approaches. To this end, we leverage LRE09 data to define a deployment evaluation protocol that involves modeled and test language groups with partial overlap. In contrast to prior evaluation protocols, this approach enables some enrolled models that correspond to unseen test languages to affect performance. We also benchmark these techniques on a held-out dataset based on the 14-language BABEL data released as part of the NIST 2016 Speaker Recognition Evaluation (SRE).

This paper is laid out as follows: Section 2 provides background on current trends in language recognition. Section 3 relates these trends to the practical application of the technology and proposes to apply binary objective calibration functions to the task. Section 4.1 defines the experiment protocol, and Section 5 gives the results and analysis.

2. Current Trends in Language Recognition

Figure 1 represents a typical multi-class calibration system that has been in many recent research publications for both closed and open-set language recognition. The main application of

This material is based upon work supported by the Defense Advanced Research Projects Agency (DARPA) under Contract No. HR0011-15-C-0037. The views, opinions, and/or findings contained in this article are those of the authors and should not be interpreted as representing the official views or policies of the Department of Defense or the U.S. Government.

http://dx.doi.org/10.21437/Interspeech.2017-530

this framework has been public language recognition benchmarks such as those hosted by NIST [9, 10] as well as programs such as DARPA RATS [1]. Bottleneck features extracted from a phonetically-aware DNN continue to provide impressive performance in recent language recognition evaluations when coupled with an i-vector extractor [3]. An i-vector extractor condenses a variable length audio file into a finite length vector of around 400 dimensions by using a universal background model and factor analysis [11]. Readers are directed to the indicated references for more information on the bottleneck feature and i-vector extraction processes.

Several different classifiers or modeling approaches (i.e., backends) leveraging i-vectors have been proposed including Support Vector Machines [12], the Gaussian backend (GB) [12], and the Adaptive Gaussian Backend [13]. In recent evaluations, including the NIST LRE'15, the Gaussian backend (GB) was predominantly used, and, since the scope of this article is concerned with the process of calibration, we constrain our analysis to likelihoods generated from the GB. The GB models each language as a Gaussian distribution with a language-dependent mean and a covariance matrix shared across all languages.

Open-set language recognition implies that some test samples will be spoken in a language not specifically targeted by the system. These test samples come from what is termed out-of-set (OOS) languages. Consequently, to cope with these OOS language tests, system developers tend to include in the GB an OOS language Gaussian model with mean given by the average mean of all training languages including those languages that the system is not tasked to detect [2, 14] and covariance given by the sum of the between- and the within-class covariances.

Scoring the GB involves extracting an i-vector from a test sample and evaluating the likelihood that it originated from each of the language detectors enrolled in the GB. These likelihoods are then typically subject to multi-class calibration followed by conversion to detection likelihood ratios (LLRs).

2.1. Multi-class calibration and detection LLRs

Multi-class calibration (MC) [2] aims to transform a set of scores computed by a system into proper likelihoods. Given a set of mutually exclusive and exhaustive hypotheses $\{H_1, \ldots, H_N\}$ about the language of a certain test segment t, and a corresponding score vector $s(t) = [s_1(t) \ldots s_N(t)]^T$, the goal of MC is to convert this score vector into a vector of relative log-likelihoods $\lambda(t)$, where $\lambda_i(t) = \log p(s(t)|H_i) + \beta$, with β being an arbitrary constant that does not affect the outcome of the final classifier. In our case, $s(t)$ is the vector of likelihoods with respect to each detector in the Gaussian backend. The transformation from s to λ is assumed to have the following form:

$$\lambda_i(t) = \alpha s_i(t) + \gamma_i \tag{1}$$

where α is a scalar shared for all languages. The parameters α and γ_i, for i between 1 and N, are estimated to minimize a multi-class Cllr objective given by

$$\text{Cllr}^{\text{MC}} = -\frac{1}{N}\sum_{i=1}^{N}\frac{1}{T_i}\sum_{t|\text{class}(t)=i}\log P_i(t) \tag{2}$$

where T_i is the number of samples from class i, and P_i is the posterior of class i given the sample t, which can be computed from λ using Bayes rule and a set of priors for each language π_i (flat priors assumed in this work) as

$$P_i(t) = \frac{\pi_i e^{\lambda_i(t)}}{\sum_{j=1}^{N}\pi_j e^{\lambda_j(t)}}. \tag{3}$$

Given the set of calibrated likelihoods λ_i for each of the N hypotheses, detection LLRs can be computed as

$$\lambda_i^{\text{det}}(t) = \log\frac{P(s(t)|H_i)}{P(s(t)|\hat{H}_i)} = \log\frac{(1-\pi_i)P_i(t)}{\pi_i\sum_{j\neq i}P_j(t)} \tag{4}$$

where $\hat{H}_i$ stands for "not H_i".

3. Binary Calibration for Language Recognition

The current framework detailed above jointly models languages such that information is shared across models. There exists two shortcomings of MC to open-set language recognition; it assumes exhaustive language detectors covering the entire language space of deployment, and it optimizes an objective that is not the one we are concerned with in the current context. Hence, we will explore directly optimizing the objective of interest, which is the binary Cllr for the detection LLRs.

Binary calibration (BC) as traditionally used in speaker recognition aims to convert scores into proper (detection, in our case) LLRs [8]. As in the case of MC, it assumes that the calibrated output is given by a linear function of the input scores. We consider two cases: a global BC and a language-dependent BC (LDBC). In the latter case, the functional form of the calibrated LLRs is identical to (1) when α is not shared across languages[1], while in the former case, the αs and γs are shared across all languages.

While the functional form of the calibration output is the same as in MC, the difference between the two calibration methods lies in the objective function. While MC optimizes a multi-class Cllr, LDBC optimizes a binary Cllr [2], where rather than N hypotheses (one for each language), two hypotheses are considered for the class i model: H_1^i = "the test sample and the detector i are the same language" and $H_2^i = \hat{H}_1^i$ = "the test sample and the detector i are different languages." Another difference between MC and BC is that the output of the calibrator is assumed to be an LLR rather than relative log-likelihoods. Hence, $\lambda_i(t) = \log\frac{P(s(t)|H_1^i)}{P(s(t)|H_2^i)}$.

The binary Cllr used as objective function during LDBC training is given by Equation (2) when $N = 2$, and the two classes are H_1^i and H_2^i. In that case, the posterior in (2) depends on the specific class i being used as a detector and is given by

$$P_1^i(t) = \frac{\pi_{\text{tar}}}{\pi_{\text{tar}} + (1-\pi_{\text{tar}})e^{-\lambda_i(t)}} \tag{5}$$

and $P_2^i(t) = 1 - P_1^i(t)$. The prior for H_1^i, π_{tar} is generally taken to be 0.5. In the case of global BC, all detectors are pooled together in the same calibration model, so the sum over t in (2) becomes a sum over t and i (the classes are still two: same detector and different detector).

Note that since λ, which is meant to be a proper LLR, is a linear function of the scores, one should feed the calibration process with scores that can be transformed into proper LLRs by a linear function. It is unreasonable to assume that the raw likelihoods from the GB will satisfy this requirement. On the other hand, a simple conversion from the GB likelihoods to (potentially miscalibrated) detection LLRs using Equation (4) works quite well in our experiments. This differs from the multi-class application in which this conversion process is applied *after* calibration (as depicted in Figure 1).

[1] In our experiments, constraining α to be shared across languages provided better performance, particularly with limited training data.

Figure 2: *LRE09 Deployment protocol defined for this work.*

Binary calibration is most effective when the same- and different-language score distributions can be modeled by the same two Gaussians irrespective of the language detector. Empirically, however, these distributions were found not to align particularly well. It was for this reason that we proposed the language-dependent binary calibration (LDBC) models. From a practical point, this is done by generating trial scores from all training i-vectors by using a single language detector before labeling same- and different-language scores and training the binary calibration model for that detector. The same process is repeated for each language detector in the system. The score distributions in this LDBC case should conform more readily to the Gaussian assumptions of the model due to variability existing only on the test side of the comparison as opposed to both detector and test variability in the BC case. A risk associated with this approach is the potential to overfit to under-trained language detectors.

4. Experimental Protocol

We define a new division of LRE09 development data for the testing of application-based language recognition systems and define the system and metrics being evaluated.

4.1. LRE09 Deployment Protocol

We define dataset splits using LRE09 data to simulate the common deployment scenario of language recognition systems. In practice, there exists three classes of languages:

1. modeled languages observed during testing
2. modeled languages **not** observed during testing
3. languages not modeled but observed during testing

To fit this scenario, we defined a protocol using LRE09 development and test data [9]. For training and enrollment of languages, 36 of the 49 original LRE09 development languages were selected. For testing, we used all 39 labeled languages in the 10 second open-set LRE09 test set [9], referred to as the *All* test set. The change from the original training set resulted in 20 languages being common to both the training and test sets (referred to as the *Seen* subset), 19 languages tested that were unknown to the system, and 16 languages modeled by the system without test samples (see Figure 2).

4.2. System, metrics and other data

The following experiments all used the same i-vector extractor, one that was trained on the 36 languages known to the system. This extractor leveraged bottleneck features extracted from a DNN trained to discriminate English senones by using 40 Mel frequency filter bank outputs stacked to have 15 frame context. For more details on the DNN, readers are referred to [4]. Bottleneck features of 80D were extracted before training a 1024 component Universal Background Model (UBM) and a 400 dimensional i-vector extractor. The GB was trained by using all 36 languages after first pre-processing the i-vectors with linear discriminant analysis (LDA) of 35 dimensions, followed by length normalization and mean normalization.

Aside from the LRE09 data, we benchmarked on the 14-language BABEL data released as part of SRE'16 [15]. The files were truncated to include only 10 seconds of dense speech. Only three of these languages were represented in the defined target languages thus the BABEL set was not only mismatched but included a high proportion of OOS test samples.

Table 1: *Benchmark of LRE09 Deployment protocol with 10 second tests using three methods of calibration. Results on mismatched BABEL data (10 second segments) are also reported.*

	LRE 09 Deployment				BABEL			
	Seen		All		Seen		All	
	Cllr	EER	Cllr	EER	Cllr	EER	Cllr	EER
MC	.133	**3.3%**	**.181**	**4.2%**	**.214**	**5.9%**	**.313**	**8.1%**
llk-BC	.170	4.0%	.193	4.6%	.359	9.3%	.385	9.9%
BC	**.126**	**3.3%**	.184	4.4%	.246	7.5%	.376	9.9%
llk-LDBC	.180	3.9%	.204	4.4%	.394	7.5%	.433	8.9%
LDBC	.177	3.7%	.234	4.6%	.245	6.6%	.374	9.6%

In this study, we focus on Cllr to measure both discriminative power and calibration performance, thus avoiding the need to define cost parameters and a detection threshold for a particular application. We also report equal error rate (EER) from pooled language trials where a trial is a comparison of a test sample to a single language detector. Note that in our context we are measuring binary Cllr and not multi-class Cllr as defined in [2].

5. Results

In this section we benchmark and analyze each of the described calibration methods on the Seen and All subsets of the LRE09 deployment protocol and BABEL datasets.

5.1. Multi-class vs binary calibration methods

The 36 target languages were used to estimate all system model parameters. During testing, 39 languages were observed, 20 of which were modeled by the system. We breakdown results in terms of all 39 languages (All tests) or restricted to the 20 languages known to the system (Seen tests). Results in terms of Cllr and EER are presented in Table 1 along with the metrics from the mismatched BABEL dataset for which the Seen and All tests consist of speech from 3 and 14 languages, respectively.

As might be expected, the results in Table 1 show that worse performance was obtained with the larger All test set compared to the Seen tests, irrespective of calibration methods and dataset combination. The llk-BC and llk-LDBC approaches do not leverage detection LLR conversion prior to calibration and are based on the GB LLKs directly. The comparison of MC and llk-BC illustrates the advantage of the MC objective leveraging the information across detectors to form an LLR for each detector. The BC and LDBC systems additionally leverage the information from other detectors to convert scores to detection LLRs *prior* to calibrating. This brings considerable gain to these systems, with BC offering performance comparable to traditional MC on LRE09. However, on the held-out BABEL dataset, BC did not maintain its gain over MC, indicating that MC was able to generalize better to the unseen conditions of BABEL. LDBC offered similar performance to BC on the BABEL dataset which differs from the trend observed on LRE09.

5.2. Modeling of OOS languages

In language recognition evaluations such as those hosted by NIST, an open-set task typically involves knowledge of a set of target languages, with other languages available when training

Table 2: *Utilizing the information of the 16 enrolled languages not observed during testing to model OOS languages in the All test cases of LRE09 Deployment and BABEL.*

	OOS	MC		BC		LDBC	
	model	Cllr	EER	Cllr	EER	Cllr	EER
LRE09	individual	**.184**	**4.3%**	**.194**	**4.6%**	**.235**	**4.8%**
	pooled	.211	4.8%	.218	5.0%	.267	5.0%
	ignored	.225	5.2%	.234	5.6%	.281	5.7%
BABEL	individual	**.348**	**9.8%**	**.433**	**12.0%**	**.425**	11.6%
	pooled	.384	10.7%	.475	12.1%	.447	**10.9%**
	ignored	.429	11.9%	.506	13.8%	.505	12.8%

the system to leverage in the modeling of OOS languages. The intention of OOS modeling strategies is to absorb test languages not targeted by the system in the open-set task. In practice, an OOS class is difficult to define without knowing which of the available target languages (36 in the current context) will either NOT be observed or not targeted during system use. The question then arises as to whether this information (i.e., subsetting the target languages) is valuable if the system could be reconfigured once deployed for alternate language groups. To answer this question, we leveraged knowledge of the 16 unobserved test languages to model the OOS language classes using several different approaches. We maintained the same number of comparable trials by removing from metric calculation any detection scores from the OOS language detectors after their information was utilized in detection LLR conversion and calibration.

In the previous section, the 16 unobserved training languages were modeled as *individual* target detectors, as were their contributions to the detection LLR and calibration schema. A common alternative is the enrollment of a single *pooled* OOS language model compensating for the additional covariance of the OOS model, by using both within and between covariance as in [2]. Lastly, the knowledge of OOS languages could be *ignored*, enabling only the tested target languages to be enrolled and leveraged in calibration and detection LLR conversion. We evaluated each of these methods on the All test sets for LRE09 and BABEL data with the results presented in Table 2. First, modeling the unobserved languages individually clearly provides the best performance across datasets and calibration methods almost consistently. The pooled method, commonly used in NIST-style evaluations, provides the second-best approach with a drop of 7–13% in all but the Babel LDBC case. One hypothesis for this finding is that the OOS languages being modeled do not overlap with the unknown languages in the test sets. This differs from the overlap observed in the LRE09 data, in which 10 of the 16 OOS training languages are common to the training and test OOS language sets. Finally, the alternate technique of ignoring the OOS data after obtaining GB covariance estimates consistently provided the worst performance.

5.3. Data balance and limitations

The LRE09 Deployment protocol consists of over 50k training segments for the 36 enrolled language detectors. However, the training data is not balanced across languages, varying from 100 audio segments to more than 8000, and the duration of these segments is not uniform. The potential issue of language bias arises with the application of binary calibration methods to language recognition. For instance, BC learns from a single pooled set of "same-language" trials that will currently be dominated by the language with the most training data. In the following experiments, we explore how sensitive each of the calibration approaches are to language bias in terms of the number of training segments per language and duration of calibration segments.

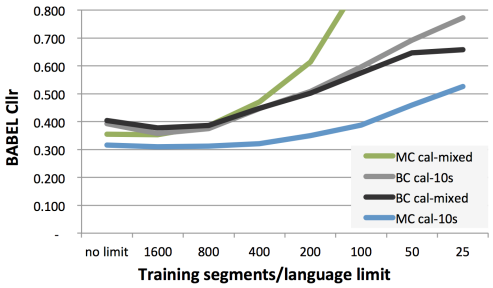

Figure 3: *Limiting the number of backend training audio segments per language, and the duration of calibration segments to observe calibration robustness in the All test condition.*

Experiments were performed on the All condition of the Babel dataset, using both mixed duration and 10 second calibration segments for the MC and BC approaches. LDBC did not offer reasonable Cllr when limiting training segments, likely due to the inherent miscalibration of score distributions from the language-dependent calibration models each learning parameters from only a handful of target trial scores. The plot in Figure 3 demonstrates that when using of all the training data per language (left side of each plot), all calibration options offer Cllr performance within 25% relative to one another. The limitation of training segments to 1600 or 800 provided better performance than using of all data for the BC approach indicating that the approach may be more sensitive to language balance than MC. Reducing the training segments to 200 or fewer makes the differences between the calibration approaches become more distinct. The MC approach significantly degrades in calibration performance as training data becomes scarce if calibration data is of mix durations. In contrast, restricting calibration segments to 10 seconds (to match test duration) significantly improved MC as training segments were reduced, despite being evaluated on the mismatched BABEL test set. The BC approach did not show sensitivity to calibration segment duration. These findings suggest that BC is sensitive to language imbalance while MC is sensitive to calibration training data from mixed durations. These trends were also validated on different duration tests including the 3 and 30 second condition of LRE09 (results not shown due to space).

6. Conclusions

This work explored applying calibration techniques with a binary objective function to the task of language recognition, including both global binary and language-dependent binary calibration. Our motivation came from the difficulty in fulfilling the assumptions of the widely adopted multi-class calibration objective in actual practice. Specifically, a target criterion not optimized for the detection task and, as suggested by the results in Section 5.2, the assumption of exhaustive detectors may not be fulfilled when train and test OOS languages do not overlap. Both multi-class and binary calibration techniques were evaluated using a system trained with a newly defined deployment protocol for LRE09 data and evaluation on LRE09 and BABEL test data. The global binary approach (BC) was shown to be comparable to the widespread multi-class approach when the train and test conditions were matched, and it offered some robustness in the context of mismatched calibration duration. Future work will investigate the application of widespread speaker recognition techniques such as score normalization to better cope with mismatched conditions in the context of binary calibration.

7. References

[1] Robust automatic transcription of speech (RATS). http://www.darpa.mil/program/robust-atuomatic-transcription-of-speech.

[2] N. Brummer and D. Van Leeuwen, "On calibration of language recognition scores," in *Odyssey: Speaker and Language Recognition Workshop*, 2006, pp. 1–8.

[3] P. Matejka, L. Zhang, T. Ng, S. Mallidi, O. Glembek, J. Ma, and B. Zhang, "Neural network bottleneck features for language identification," in *Proc. Speaker Odyssey*, 2014.

[4] L. Ferrer, Y. Lei, M. McLaren, and N. Scheffer, "Study of senone-based deep neural network approaches for spoken language recognition," *IEEE Transactions on Acoustics Speech and Signal Processing*, 2015, accepted for publication.

[5] R. Fér, P. Matějka, F. Grézl, O. Plchot, and J. Černockỳ, "Multilingual bottleneck features for language recognition," in *Sixteenth Annual Conference of the International Speech Communication Association*, 2015.

[6] M. McLaren, L. Ferrer, and A. Lawson, "Exploring the role of phonetic bottleneck features for speaker and language recognition," in *Proc. ICASSP*, 2016.

[7] F. Richardson, D. Reynolds, and N. Dehak, "A unified deep neural network for speaker and language recognition," in *Proc. Interspeech*, 2015.

[8] N. Brümmer and J. Du Preez, "Application-independent evaluation of speaker detection," *Computer Speech & Language*, vol. 20, no. 2, pp. 230–275, 2006.

[9] *The 2009 NIST language recognition evaluation plan*, 2009, http://www.itl.nist.gov/iad/mig/tests/lre/2009/.

[10] *The 2015 NIST language recognition evaluation plan*, 2015, http://www.nist.gov/itl/iad/mig/upload/LRE15_EvalPlan_v23.pdf.

[11] N. Dehak, P. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet, "Front-end factor analysis for speaker verification," *IEEE Trans. on Speech and Audio Processing*, vol. 19, pp. 788–798, 2011.

[12] 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 *Proc. Interspeech*, 2003.

[13] M. McLaren, A. Lawson, Y. Lei, and N. Scheffer, "Adaptive gaussian backend for robust language identification." in *Proc. Interspeech*, 2013, pp. 84–88.

[14] N. Brummer, L. Burget, O. Glembek, V. Hubeika, Z. Jancik, M. Karafiát, P. Matejka, T. Mikolov, O. Plchot, and A. Strasheim, "But-agnitio system description for nist language recognition evaluation 2009," in *Proceedings NIST 2009 Language Recognition Evaluation Workshop*, 2009, pp. 1–7.

[15] *The NIST Year 2016 Speaker Recognition Evaluation Plan*, 2016, https://www.nist.gov/sites/default/files/documents/2016/10/07/sre16_eval_plan_v1.3.pdf.