SPOKEN LANGUAGE RECOGNITION WITH CLUSTER-BASED MODELING

Stanisław Kacprzak, Magdalena Rybicka, Konrad Kowalczyk

Faculty of Computer Science, Electronics and Telecommunications
AGH University of Science and Technology, 30-059 Krakow, Poland
{skacprza, mrybicka, konrad.kowalczyk}@agh.edu.pl

ABSTRACT

In this study, we analyze the incorporation of cluster-based modeling into the language recognition systems, in which a single utterance is represented as an embedding, deploying widely used i-vectors and x-vectors. We compare the results obtained with a Cosine Distance Scoring, Gaussian Mixture Model, Logistic Regression, and the Mixture of von Misses-Fisher distributions with the classifiers based on the proposed approach which incorporates cluster-based sub-models. Experimental evaluation is performed on the i-vector embeddings from the NIST 2015 language recognition i-vector machine learning challenge and the x-vector embeddings from the Oriental Language Recognition 2020 Challenge (AP20-OLR). The experimental results clearly show that the proposed approach combined with discriminatively trained Logistic Regression classifier achieves notable improvements over the baseline systems, i.e., those without language sub-models, and that our approach is competitive to other systems reported in the literature.

***Index Terms*—** language recognition, AP20-OLR challenge, i-vectors, x-vectors, clustering

1. INTRODUCTION

In this work, we address the problem of language recognition. In particular, we analyze the application of unsupervised clustering algorithms in the embedding space to create multiple independent language models. The incorporation of language sub-models is envisioned to improve the performance of the language recognition systems, especially when dealing with a large number of languages or closely related languages.

The well-known embeddings of speech utterances are i-vectors [1] and x-vectors [2]. Although x-vectors are predominantly applied in recent language recognition systems, i-vectors remain in the interest of the scientific community [3]. Furthermore, by providing complementary information, they are often used for system fusion.

To our knowledge, a similar approach to modeling languages with unsupervised clusters has previously been used only in [4], where an agglomerative hierarchical clustering with the cosine distance combined with sPLDA were applied, and in [5] where a two-component Gaussian Mixture Model (GMM) achieved a slight improvement compared with a single-component model, when trained with a large amount of data. Our initial experiments with the concept were presented in [6].

When modeling the language with a mixture model, clusters are weighted together by the priors. This could be considered unwanted if language clusters were treated as separate sub-classes. If one cluster had small cardinality compared to the rest of the data, it would be reflected in its weight, and as a result, its contribution to the entire model would be small. On the other hand, treating a cluster as a separate sub-class neglects the prior, which can be desirable if the language clusters represent dialects or channels. Furthermore, as stated in [7], the prior of the class cannot be supplied by the technology. Since the embedding space contains relevant information that can be modeled by the underlying parameterization, the existence of clusters can be connected with the nature of a given language. Then their cardinalities should be reflected in the priors.

One of the first attempts to combine supervised and unsupervised learning to find sub-classes was presented in [8]. The author argued that engineer-assigned class labels are not always producing homogeneous classes, which is required by some classification methods to work correctly. We hypothesize that this is also the case for language recognition because heterogeneous language classes can be produced due to specific language nature. For instance, Chomsky in [9] points out that some of the German dialects are more similar to Dutch than to other German dialects. Evidence of channel and gender distortions observed in x-vector embeddings was reported in [10].

Other approaches to deal with multimodal distributions in language data include local Fisher discriminant analysis (LFDA) [11] and nearest neighbor discriminant analysis [12]. Clustering algorithms were used in [13] to find clusters of close languages, which allow to create hierarchical language identification system.

In this work, the proposed incorporation of cluster-based language modeling is evaluated on two data sets. The first data set is from the NIST 2015 language recognition i-vector machine learning challenge (NIST i-vector LRE) [14]. It contains a large number of 50 languages. The second data set is from the recent Oriental Language Recognition Challenge 2020 (AP20-OLR) [15], and it contains a small number of oriental languages recorded with different channels. We focus only on modeling languages present in the training data set, thus performing experiments only in the closed-set scenario, although clustering of unlabeled data for out-of-set languages modeling can be beneficial [16]. The main difference to the approach presented in [4] is the incorporation of clusters into a discriminative classifier. After describing a range of classifiers (Sec. 2), we present the proposed approach (Sec. 3), followed by the description of data sets, experiments, result discussion, and conclusions (Secs. 4, 5, 6).

2. CLASSIFIERS

2.1. Cosine Distance Scoring

Cosine Distance Scoring (CDS) is a simple yet powerful way of scoring an unknown speech utterance, often applied in the i-vector space.

This research was supported by the Foundation for Polish Science under grant number First TEAM/2017-3/23 (POIR.04.04.00-00-3FC4/17-00) which is co-financed by the European Union, and in part by PLGrid Infrastructure.

It consists of calculating the dot product of the length-normalized test and target embeddings, the latter represents the language. It is commonly used as a baseline or sub-system.

2.2. Gaussian Mixture Model

Gaussian Mixture Model (GMM) is defined as

$$p(x|w_k, \mu_k, \Sigma_k) = \sum_{k=1}^{K} w_k \, \mathcal{N}\left(x|\mu_k, \Sigma_k\right) \,, \tag{1}$$

where μ_k and Σ_k are the mean and covariance of the k-th distribution, K is the number of Gaussian components, and w_k is the mixing weight which satisfies the constraint $\sum_{k=1}^{K} w_k = 1$. In this work we use diagonal Σ_k, which is shared across all components of the model. We also consider a special case of single-component models with full covariance matrix shared across models for all languages. We will refer to this model as a Gaussian Classifier (GC).

2.3. Mixtures of von Mises-Fisher Distributions

Since the GMM utilizes Euclidean distances, if length normalization is applied, which is a common practice, the embeddings lie on a hypersphere. In this case, better suited mathematical models exist [17]. The Mixture Model of von Mises-Fisher distributions (vMF-MM) provide a way to model directional data (i.e., probability distribution on a sphere). The von Mises-Fisher distribution [18] of d-dimensional data is defined as

$$f(x, \mu, \kappa) = \frac{\kappa^{d/2-1}}{(2\pi)^{d/2}\, I_{d/2-1}(\kappa)} e^{\kappa\mu^\top x} \tag{2}$$

where $\kappa \geq 0$, $\|\mu\| = 1$, and $I_{d/2-1}$ is a modified Bessel function of the first kind of order $d/2 - 1$. Parameter μ represents the mean direction and κ is a concentration parameter. Then the vMF-MM is a weighted sum of von Mises-Fiher distributions

$$p(x) = \sum_{k=1}^{K} w_k \; f(x, \mu_k, \kappa_k), \tag{3}$$

where $\sum_{k=1}^{K} w_k = 1$. Parameters w_k and μ_k are estimated from training data, commonly with an EM algorithm [19], but the estimation of κ_k has no analytical solution, and hence several methods exist in the literature for calculating its approximation.

2.4. Logistic Regression

Logistic Regression (LR) is a binary classifier that solves an optimization problem defined by

$$\min_{w,b} \frac{1}{2} w^\top w + C \sum_{i=1}^{N} \log(1 + e^{-y_i(w^\top x_i + b)}) \,, \tag{4}$$

where N denotes the number of training data points, y_i takes either 1 or -1 values depending if the i-th training point belongs to the positive or negative class, w and b are parameters obtained during the training and C is a penalty parameter. This formulation is an ℓ_2-norm regularized logistic regression. In this work, we follow the popular technique of using multiple binary classifiers trained with *one-vs-rest* strategy.

3. LANGUAGE CLUSTER-BASED MODELING

In this section, we introduce the proposed cluster-based modeling of languages. We aim to discover sub-classes in the data that can be used for creating multiple models for each class.

Note that the ultimate goal is to maximize the language recognition performance rather than to find the best clustering. One could expect that these two are dependent, but it does not have to be the case. In search of the best clustering strategy, in preliminary experiments, we found out that a constant number of clusters for all languages works better than the estimation of the optimum number of clusters according to the internal clustering quality measures. We have also experimented [20] with a greedy heuristic (because checking all combinations of clustering for a large number of languages is an intractable problem) that measured the impact of different clustering on the classification results, and also concluded that a constant number of clusters is a better choice. These observations are well in line with the results obtained in [21] where the clustering using the x-means algorithm (that determines the number of clusters for each class) did not outperform the k-means algorithm with a constant K for all classes. This suggests that even when there are no evident clusters in the class data, using K sub-models increases linear decision hyperplanes and approximation of nonlinear separation boundaries. We hypothesize that this effect may be observed also in our experiments since some languages lie very close to each other (with possible overlapping) in the embedding space and splitting a single (unimodal) language cluster into smaller parts could improve the separation between classes.

3.1. Clustering algorithm

For clustering, we use the spherical k-means algorithm. In its standard form, it is based on a Euclidean distance and it is one of the most popular clustering algorithms, but if cosine distance $\mathrm{d}_{\cos}$ is used instead, it is known as spherical k-means [22]. The algorithm's only parameter is K, which is the number of clusters. In the first step, the algorithm assigns initial values of the centroids m_k for each cluster. Then, at each step t, every observation x is assigned to one of the clusters S_k^t with the closest centroid m_k^t, and new cluster centers are calculated as

$$m_k^{t+1} = \sum_{x \in S_k^t} \frac{x}{\|x\|} \,. \tag{5}$$

The algorithm defined this way finds partitioning $S = \{S_1, \ldots, S_k\}$ that minimizes within-cluster sums of point-to-centroid cosine distances

$$\arg\min_{S} \sum_{k=1}^{K} \sum_{x \in S_k} \mathrm{d}_{\cos}(x, m_k) \,. \tag{6}$$

For initialization of the algorithm during parameter search on the validation set, we used k-means++ [23] with replication set to five. However, when calculating the presented results, deterministic initialization described in [24] was used, which calculates the well-distributed seeds across the input space. In [20], we experimented on NIST i-vector LRE with other clustering algorithms such as mean shift and agglomerative hierarchical clustering, but we found that clusters determined by k-means, in general, provided better results for the task.

3.2. Integration of cluster-based models with classifiers

Since GMM and vMF-MM represent soft clustering algorithms, we combine spherical k-means clustering, to produce separate language models, with CDS and LR classifiers.

Incorporating cluster-based modeling is straightforward for the CDS classifier. We cluster samples of every language and calculate the centroids of each cluster. Then, for each test embedding, we calculate the distance to each of these centroids. The predicted language corresponds to the language of the closest centroid.

To incorporate cluster-based modeling into a logistic regression, we modify the training algorithm. Using *one-vs-rest* strategy, we have a separate classifier for each of the clusters, so it is crucial, that during the training, we exclude the samples from the language of a positive class that belongs to other clusters of that language, which should not be considered as negative examples (the rest in *one-vs-rest*). The main reason for that is to neglect miss-classification errors between the clusters of the same language.

4. DATA SETS AND EVALUATION SETUP

4.1. NIST i-vector LRE 2015

One of the data sets used in experiments is a set of i-vectors from the NIST 2015 language recognition i-vector machine learning challenge (NIST i-vector LRE) [14]. The speech data for the challenge was shared in the form of already computed i-vectors of dimensionality $d = 400$. The computation of i-vectors followed the scheme presented in [1]. The task in NIST i-vector LRE was an open-set language recognition, however, in our experiments, we deal only with the closed-set case. For that reason, similarly to [25–28], we use the entire training data (15000 i-vectors) but the reduced test data (5000 i-vectors without out-of-set samples). Furthermore, we did not use the development set which contains unlabeled data. Both training and test sets contain uniformly distributed data of 50 languages.

4.2. AP20-OLR x-vectors

As the second data set, we incorporated x-vectors extracted with PyTorch-based baseline embedding extractor [2,29] provided during the Oriental Language Recognition 2020 Challenge (AP20-OLR) [30]. As input features, we used Mel-Frequency Cepstral Coefficients (MFCC) providing 20 features that were concatenated with additionally extracted 3-dimensional pitch features. Speech frames were detected by Kaldi [31] energy-based Voice Activity Detector with a threshold set to 5.0 and mean-normalized with a 3 s window. The x-vector system was represented by Extended Time Delay Neural Network (ETDNN) [29]. Embeddings were extracted from the first layer after the pooling operation producing a vector of 512 length. The training procedure was similar to that of the baseline system [30].

As training data, we incorporated the recordings provided in the previous OLR challenges, namely AP16-OL7 [32], AP17-OL3 and AP17-OLR-test [33] subsets. Data set included in total 10 languages (Cantonese, Mandarin, Indonesian, Japanese, Russian, Korean, Vietnamese, Kazakh, Tibetan, and Uyghur), all obtained from mobile channels. Data augmentation involved speed perturbation with a factor of 0.9 and 1.1, which increased the amount of training data three times. Then, the volume perturbation with a randomly selected factor from 0.125-2.0 was applied to all recordings. The systems were evaluated on the data set for Task 1 of the AP20-OLR challenge, which represented a cross-channel language identification task. The set covered six languages, namely Cantonese, Indonesian, Japanese, Russian, Korean and Vietnamese. The development set for this task consisted of a subset of AP19-OLR-test for the cross-channel task [34], which included languages: Tibetan, Uyghur, Japanese, Russian, Vietnamese, and Mandarin.

4.3. Evaluation metric

In our experiments, we measure system recognition performance with $Cost$ which is a classification error averaged across all language classes presented in percent. This is equivalent to the measure used in the evaluation system from the NIST i-vector challenge [14], except that we omit the cost of misclassification of out-of-set languages. To provide means of comparison of our results with other works, especially those from the AP20-OLR challenge, we calculate also the standard measure of average decision cost function C_{avg} as defined in [35]. This measure calculates the average cost of the system in a scenario where it is used as a language detector for each language.

5. EXPERIMENTS AND RESULT DISCUSSION

5.1. Experimental setup

In all systems, except for the baselines (which were provided by the organizers), we performed centering, applied WCCN [36] or LDA transformations, and performed length-normalization of embeddings. In the case of the NIST i-vector LRE experiments, only the GMM and CDS systems performed better with dimensions reduced to 49 by the LDA transformation, while the rest of the systems used WCCN. Hyperparameters of classifiers were tuned with the use of 5-fold cross-validation, and we experimented with values of K up to 5. All AP-OLR20 systems used x-vectors with dimensions reduced to 9 by the LDA transformation (we used data for all 10 languages for LDA transformation, but created models only for 6 target languages). Hyperparameters of classifiers were tuned with the use of the validation set (containing 3 target languages) extracted from the development set, but the entire development set was added to the training set for final evaluation. For this data set, we experimented with values of K up to 100 (because there are more samples and fewer languages than in the NIST i-vector LRE data set).

Baseline systems were based on CDS for the NIST i-vector LRE and LR for AP-OLR20, respectively. In experiments with vMF-MM, our approach differs from the one presented in [37]. We trained separate models using single language data for each model and use an approximation of κ presented in [19]. In [20], we tested different approaches of estimating the optimum number of components on the NIST i-vector LRE data set, and the best results were obtained when the maximum number of mixture components was chosen using the Bayesian information criterion (similarly to [37]) but constrained to a maximum $K_{\max}$. Regarding implementation, we used the FoCal [38] toolkit for GC and LIBLINEAR [39] library for LR.

5.2. Results and discussion

Table 1 presents the results for all studied classifiers on both data sets. Comparing four types of studied classifiers, we can clearly observe that the best results are obtained by the LR, which is a discriminative classifier. The vMF-MM model (with $K_{\max}$ = 3) achieves the best results from generative models on the NIST i-vector LRE, but on AP20-OLR it is reduced to the single-component models and poor performance is observed. GMM with two components outperforms the GMM model with a single component and GC model on NIST i-vector LRE. However, GMM performs poorly compared with GC on AP20-OLR.

Table 1. Language recognition results on NIST i-vector LRE data set and x-vector system for AP20 OLR Task 1: cross-channel LID.

	NIST LRE (i-vectors)		AP20-OLR (x-vectors)	
System	$Cost$	$100 \cdot C_{avg}$	$Cost$	$100 \cdot C_{avg}$
Challenge baseline	20.96	10.69	-	13.21
CDS	17.18	8.77	20.20	12.12
CDS (with clustering)	15.82	8.07	20.06	12.03
GC	16.84	8.59	19.93	11.96
GMM (K=1)	17.64	9.00	25.66	12.86
GMM (K=2)	16.56	8.45	27.27	12.33
vMF-MM	15.78	8.05	22.83	13.70
LR	15.32	7.82	16.89	10.13
LR (with clustering)	**14.80**	**7.55**	**16.08**	**9.65**

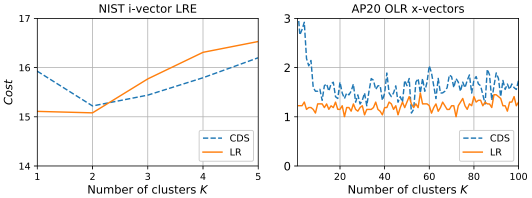

Fig. 1. Dependency of $Cost$ from the number of clusters K from cross-validation (NIST i-vector LRE) and validation (AP20 OLR x-vectors).

Using classifiers that incorporate the proposed cluster-based modeling, we can observe that the proposed approach performs notably better than its counterpart without sub-models. For all systems on NIST i-vector LRE, the optimal value of K is 2 (due to a relatively small size of samples per language). For AP-OLR20, the best validation results are obtained for K=52 for the CDS and K=72 for the LR, however, setting K=2 would allow to achieve $Cost$ of 16.26 (which is still an improvement). Figure 1 presents the results of $Cost$ for different values of K on the AP-OLR20 validation set.

In an additional experiment, we verify our approach on five closely related languages from NIST i-vector LRE: Czech, Polish, Russian, Slovak, and Ukrainian. By adding the clustering to the LR classifier, the $Cost$ is reduced from 12.80 to 11.40.

The overall relative improvement in $Cost$ for the NIST i-vector LRE data set obtained with the modified LR classifier reaches 29% compared to the baseline system and 3% over the LR system without the proposed modification. The respective values of C_{avg} for the AP20-OLR data set are 27% and 5%.

5.3. Comparision to other works

The presented results on the NIST LRE data set are competitive to those reported in the literature. As expected, using nonlinear classifier allows for obtaining better results. To our knowledge, the lowest reported closed-set $Cost$ is presented in [26], where i-vector based conditional Generative Adversarial Network (cGAN) classifier achieved 11.34%. However, another version of this system (trained with SGD optimizer) achieved 14.60, which is comparable to the result of our approach. In [40] the LR classifier outperformed the cGAN classifier on a different data set, and we speculate that its performance could be further improved with our proposed modification. Better results were reported in [28], while our system outperforms the systems presented in [25] and [27].

For the AP20-OLR data set, the best system [41] for cross-channel LID task, which was developed during the challenge, achieved $100 \cdot C_{avg}$ of 2.39 [15]. This system fully utilized the provided data by using audio transcripts for the training of multilingual bottleneck features, which we do not consider in this work. In comparison to other challenge submissions [15], the proposed cluster-based model with the LR classifier should outperform many of the submitted systems.

6. CONCLUSIONS

In this study, we analyzed the impact of language cluster-based models on the language recognition task. We modified training schemes for the LR and CDS, which enable incorporating language submodels. The proposed modifications provide consistent improvement in all performed experiments. The presented modifications to the language recognition system increase computational complexity by factor K, however, since obtained models are independent, computations can be conveniently parallelized. The proposed approach is validated on two standard databases using i-vector and x-vector embeddings.

7. REFERENCES

[1] N. Dehak, P. A. Torres-Carrasquillo, D. A. Reynolds, and R. Dehak, "Language recognition via i-vectors and dimensionality reduction." in *Interspeech*, 2011, pp. 857–860.

[2] D. Snyder, D. Garcia-Romero, A. McCree, G. Sell, D. Povey, and S. Khudanpur, "Spoken language recognition using x-vectors," in *Proc. Odyssey 2018 The Speaker and Language Recognition Workshop*, 2018, pp. 105–111.

[3] A. I. Abdurrahman and A. Zahra, "Spoken language identification using i-vectors, x-vectors, plda and logistic regression," *Bulletin of Electrical Engineering and Informatics*, vol. 10, no. 4, pp. 2237–2244, 2021.

[4] D. M. González, J. V. López, E. L. Solano, and A. O. Gimenez, "Unsupervised accent modeling for language identification," in *Advances in Speech and Language Technologies for Iberian Languages*. Springer, 2014, pp. 49–58.

[5] I. Lopez-Moreno, J. Gonzalez-Dominguez, O. Plchot, D. Martinez, J. Gonzalez-Rodriguez, and P. Moreno, "Automatic language identification using deep neural networks," in *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*. IEEE, 2014, pp. 5337–5341.

[6] S. Kacprzak, "Spoken language clustering in the i-vectors space," in *2017 International Conference on Systems, Signals and Image Processing (IWSSIP)*. IEEE, 2017, pp. 1–5.

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

[8] N. Japkowicz, "Supervised learning with unsupervised output separation," in *International conference on artificial intelligence and soft computing*, vol. 3, 2002, pp. 321–325.

[9] N. Chomsky, *Rules and Representations*, ser. Columbia Classics in Philosophy. Columbia University Press, 1980.

[10] R. Duroselle, D. Jouvet, and I. Illina, "Metric learning loss functions to reduce domain mismatch in the x-vector space for

language recognition," in *Proc. Interspeech 2020*, 2020, pp. 447–451.

[11] P. Shen, X. Lu, L. Liu, and H. Kawai, "Local Fisher discriminant analysis for spoken language identification," in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2016, pp. 5825–5829.

[12] S. O. Sadjadi, J. W. Pelecanos, and S. Ganapathy, "Nearest neighbor discriminant analysis for language recognition," in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2015, pp. 4205–4209.

[13] S. Irtza, V. Sethu, H. Bavattichalil, E. Ambikairajah, and H. Li, "A hierarchical framework for language identification," in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2016, pp. 5820–5824.

[14] "Language Recognition i-vector Machine Learning Challenge," https://www.nist.gov/document/lreivectorchallengerelv1-0pdf.

[15] J. Li *et al.*, "Oriental Language Recognition (OLR) 2020: Summary and Analysis," in *Proc. Interspeech 2021*, 2021, pp. 3251–3255.

[16] Q. Zhang and J. H. Hansen, "Training candidate selection for effective out-of-set rejection in robust open-set language identification," *The Journal of the Acoustical Society of America*, vol. 143, no. 1, pp. 418–428, 2018.

[17] O. C. Hamsici and A. M. Martinez, "Spherical-homoscedastic distributions: The equivalency of spherical and normal distributions in classification," *Journal of Machine Learning Research*, vol. 8, no. Jul, pp. 1583–1623, 2007.

[18] K. V. Mardia and P. E. Jupp, *Directional statistics*. John Wiley & Sons, 2009, vol. 494.

[19] A. Banerjee, I. S. Dhillon, J. Ghosh, and S. Sra, "Clustering on the unit hypersphere using von mises-fisher distributions," *Journal of Machine Learning Research*, vol. 6, no. Sep, pp. 1345–1382, 2005.

[20] S. Kacprzak, "Spoken language recognition in i-vector space using cluster-based modeling," Ph.D. dissertation, AGH University of Science and Technology, 2020.

[21] D. Fradkin, "Clustering inside classes improves performance of linear classifiers," in *Tools with Artificial Intelligence, 2008. ICTAI'08. 20th IEEE International Conference on*, vol. 2. IEEE, 2008, pp. 439–442.

[22] I. S. Dhillon and D. S. Modha, "Concept decompositions for large sparse text data using clustering," *Machine learning*, vol. 42, no. 1-2, pp. 143–175, 2001.

[23] D. Arthur and S. Vassilvitskii, "K-means++: The advantages of careful seeding," in *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*. Society for Industrial and Applied Mathematics, 2007, pp. 1027–1035.

[24] R. Duwairi and M. Abu-Rahmeh, "A novel approach for initializing the spherical k-means clustering algorithm," *Simulation Modelling Practice and Theory*, vol. 54, pp. 49–63, 2015.

[25] X. Lu, P. Shen, Y. Tsao, and H. Kawai, "Regularization of neural network model with distance metric learning for i-vector based spoken language identification," *Computer Speech & Language*, vol. 44, pp. 48–60, 2017.

[26] P. Shen, X. Lu, S. Li, and H. Kawai, "Conditional generative adversarial nets classifier for spoken language identification," *Proc. Interspeech 2017*, pp. 2814–2818, 2017.

[27] P. Heracleous, Y. Mohammad, K. Takai, K. Yasuda, and A. Yoneyama, "Spoken language identification based on i-vectors and conditional random fields," in *2018 14th International Wireless Communications & Mobile Computing Conference (IWCMC)*. IEEE, 2018, pp. 1443–1447.

[28] P. Heracleous, K. Takai, K. Yasuda, Y. Mohammad, and A. Yoneyama, "Comparative study on spoken language identification based on deep learning," in *2018 26th European Signal Processing Conference (EUSIPCO)*. IEEE, 2018, pp. 2265–2269.

[29] D. Snyder, D. Garcia-Romero, G. Sell, A. McCree, D. Povey, and S. Khudanpur, "Speaker recognition for multi-speaker conversations using x-vectors," in *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2019, pp. 5796–5800.

[30] Z. Li *et al.*, "AP20-OLR Challenge: Three Tasks and Their Baselines," 2020.

[31] D. Povey *et al.*, "The Kaldi speech recognition toolkit," in *IEEE 2011 Workshop on Automatic Speech Recognition and Understanding*. IEEE Signal Processing Society, Dec. 2011.

[32] D. Wang, L. Li, D. Tang, and Q. Chen, "Ap16-ol7: A multilingual database for oriental languages and a language recognition baseline," in *2016 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA)*, 2016, pp. 1–5.

[33] Z. Tang, D. Wang, Y. Chen, and Q. Chen, "AP17-OLR Challenge: Data, plan, and baseline," in *2017 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, 2017, pp. 749–753.

[34] Z. Tang, D. Wang, and L. Song, "AP19-OLR Challenge: Three tasks and their baselines," *2019 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, pp. 1917–1921, 2019.

[35] N. L. Group *et al.*, "The 2007 NIST Language Recognition Evaluation Plan (LRE07)," http://www.nist.gov/speech/tests/lang/2007/LRE07EvalPlan-v7e.pdf, 2007.

[36] A. O. Hatch and A. Stolcke, "Generalized linear kernels for one-versus-all classification: application to speaker recognition," in *Acoustics, Speech and Signal Processing, 2006. ICASSP 2006 Proceedings. 2006 IEEE International Conference on*, vol. 5. IEEE, 2006, pp. V–V.

[37] I. Lopez-Moreno, D. Ramos, J. Gonzalez-Dominguez, and J. Gonzalez-Rodriguez, "Von Mises-Fisher models in the total variability subspace for language recognition," *IEEE Signal Processing Letters*, vol. 18, no. 12, pp. 705–708, 2011.

[38] N. Brummer, "Focal multiclass toolkit," *URL: http://niko. brummer. googlepages. com/focalmulticlass*, 2014.

[39] 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.

[40] X. Miao, I. McLoughlin, S. Yao, and Y. Yan, "Improved conditional generative adversarial net classification for spoken language recognition," in *2018 IEEE Spoken Language Technology Workshop (SLT)*. IEEE, 2018, pp. 98–104.

[41] R. Duroselle, M. Sahidullah, D. Jouvet, and I. Illina, "Language recognition on unknown conditions: The LORIA-Inria-MULTISPEECH system for AP20-OLR Challenge," in *Proc. Interspeech 2021*, 2021, pp. 3256–3260.