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

Investigating Scalability in Hierarchical Language Identification System

Saad Irtza[1,2], *Vidhyasaharan Sethu*[1], *Eliathamby Ambikairajah*[1,2], *Haizhou Li*[3]
[1]School of Electrical Engineering and Telecommunications, UNSW Australia
[2] DATA61, CSIRO, Sydney, Australia
[3]National University of Singapore, Singapore
s.irtza@unsw.edu.au

Abstract

State-of-the-art language identification (LID) systems are not easily scalable to accommodate new languages. Specifically, as the number of target languages grows the error rate of these LID systems increases rapidly. This paper addresses such a challenge by adopting a hierarchical language identification (HLID) framework. We demonstrate the superior scalability of the HLID framework. In particular, HLID only requires the training of relevant nodes in a hierarchical structure instead of re-training the entire tree. Experiments conducted on a dataset that combined languages from the NIST LRE 2007, 2009, 2011 and 2015 databases show that as the number of target languages grows from 28 to 42, the performance of a single level (non-hierarchical) system deteriorates by around 11% while that of the hierarchical system only deteriorates by about 3.4% in terms of C_{avg}. Finally, experiments also suggest that SVM based systems are more scalable than GPLDA based systems.

Index Terms: language identification, hierarchical LID, scalable framework, neural networks, deep learning

1. Introduction

In the last decade, spoken language identification has gained interest from the research community to be used as an auxiliary technology for many applications e.g. speech recognition and dialogue systems [1, 2]. To integrate with these applications, LID systems should be scalable to fairly large number of languages in order to serve the wider community. The term scalability refers to an LID system's potential to accommodate the growth without significant performance degradation [3], i.e. to include new target languages/dialects with minimal re-training and performance degradation. Language identification systems can be easily developed on large scale if all the languages and its dialects speech data are available at once. Unfortunately, it is unfeasible to collect speech data to cover all languages, so we often choose to add new languages/dialects [4, 5] only in response to application needs. As such, scalability is a highly desirable trait in LID system.

The state-of-the-art LID systems rely on a total variability factor analysis (FA) approach known as the i-vector framework [6, 7]. This framework makes use of acoustic and phonotactic information. Specifically, MFCCs and phone log likelihood ratios (PLLRs) continue to be the most commonly used acoustic and phonotactic front-ends [1, 8, 9] and recently bottleneck features (BNFs) [10] have exhibited promising performances. Deep learning approaches e.g. Convolutional Neural Network (CNN), Recurrent Neural Network (RNN) and Long Short Term Memory (LSTM) have also shown competitive performance [11]. In the past few years, deep learning has also been successfully used to develop end-to-end LID systems and become popular among researchers [12]. There are several choices of classifier to be used such as Gaussian Probabilistic Linear Discriminant Analysis (GPLDA), Generative Gaussian (GG) classifiers and Support Vector Machine (SVMs) [13-15]. Most commonly the abovementioned approaches are used in single-level LID (SLID) frameworks, where all language hypotheses are treated identically [1].

Alternatively, previously proposed hierarchical frameworks organize languages in groups, to form a hierarchical tree structure [16, 17], based on the observation that it is easier to distinguish between languages that are significantly dissimilar than those having a lot of similarities [18, 19]. It has also been observed that the cues that are utilised to distinguish between languages depend on how similar they are, e.g. prosodic cues are significantly better at distinguishing between tonal and non-tonal languages than they are at distinguishing between two non-tonal languages [2]. Preliminary work on the use of hierarchical structures have been proposed as an alternative to the traditional single level structure and has shown some promising performance [18, 19]. There are several approaches to creating hierarchical structures. For example, automatic language clustering algorithms have been be used to form a hierarchical structure [18]. Alternatively, linguistic language families and phonotactic information have also formed the basis for a hierarchical structure [20].

Most of the above LID frameworks, both single level and hierarchical, rely on the i-vector framework [13, 17]. The i-vectors are extracted from a low dimensional subspace known as the total variability (TV) space, which is learned using all target languages' data [8]. In order to expand this framework to include new target languages/dialects, the TV space, i-vector extraction and language modelling modules require re-training, which seems to require extra effort especially when including a small number of languages in a system that contains many more. The computational complexity is also affected if the LID system consists of several layers of CNNs, RNNs and DNNs as it requires the re-optimization of each network. Such an LID system's performance significantly deteriorates and it is not easily scalable to accommodate new languages [21-23]. The question arises how to effectively scale an existing LID system with minimal performance deterioration and effort required. Our current work is focused on this research question and in this paper we propose a novel approach to training the hierarchical framework to accommodate new languages by introducing language cluster specific features. To the best of author's knowledge this is the first paper to address the scalability of LID systems. This work also investigates the use of a one vs. all SVM classification approach for scalable LID systems.

http://dx.doi.org/10.21437/Interspeech.2017-596

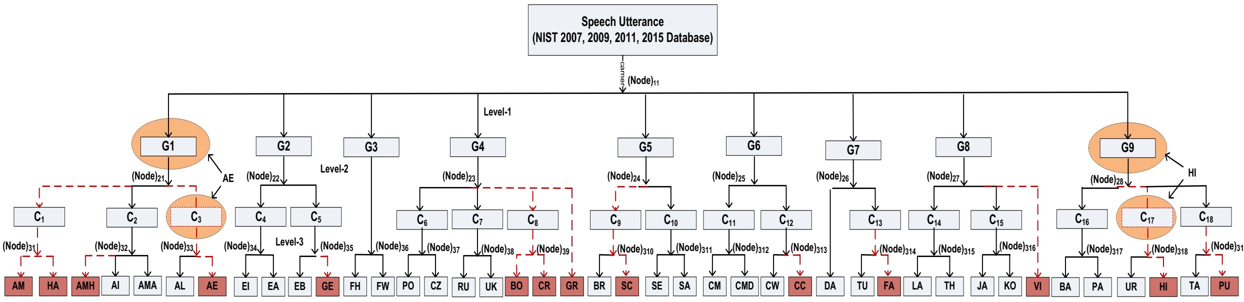

Figure 2: *Hierarchical Language Identification Framework on NIST 2007, 2009, 2011 and 2015 Database.*

2. Scalability of LID System

Most commonly in image processing, scalable systems are developed by using a hierarchical framework in which classes are organized to form a tree structure [3]. Motivated by this approach, this work investigates the scalability of an LID system in a hierarchical framework. A high-level block diagram of the hierarchical language identification framework is shown in Figure 1. In such a framework, similar languages are grouped to form language clusters at each level of hierarchy structure. Each node is designed to classify between languages or language groups associated with the respective node. Once the hierarchical structure is developed, it can be expanded to include new languages by first determining the appropriate language group at each level. Secondly, it either requires upgrading or developing a new node in the lower levels to represent the newly added languages associated with that node. For example, Figure 1 shows that after determining the language group of new languages, *F* and *G*, hierarchy (node)$_{31}$ and (node)$_{21}$ are upgraded respectively. The nodes in the upper levels may not require upgrading as they are already developed on languages similar to new language i.e. it is not essential to re-train hierarchy (node)$_{11}$ and (node)$_{22}$.

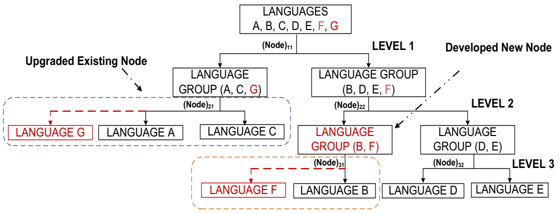

Figure 1: *Example of Hierarchical Language Identification Framework.*

3. Proposed Hierarchical Structure Training

In this work, a previously developed agglomerative hierarchical clustering algorithm is used again to form the initial hierarchical structure [14, 17]. This clustering algorithm is extended to determine the language groups of new languages. The algorithm uses pairwise similarities between languages/language groups using phonotactic and linguistic information. The similarity, $S(\cdot)$, between a language pair (L_a, L_b) is computed as per equation (1), where $a, b \in (1, N)$ and N is number of languages.

$$S(L_a, L_b) = \big(1 - K_s(L_a, L_b)\big) \times E(L_a, L_b) \qquad (1)$$

Where, $K_s(\cdot)$ is the symmetric K-divergence between phoneme probability distribution of language L_a and L_b. The symmetric K-divergence between two languages, L_a and L_b, is defined as [24]:

$$K_s(L_a, L_b) = \sum_{i=1}^{N_P} \Bigg[P(p_i|L_a) \ln\left(\frac{2P(p_i|L_a)}{P(p_i|L_a) + P(p_i|L_b)}\right) + P(p_i|L_b) \ln\left(\frac{2P(p_i|L_b)}{P(p_i|L_a) + P(p_i|L_b)}\right)\Bigg] \qquad (2)$$

Here, N_p is the number of unigrams/bigrams considered, and $P(p_i|L_j)$ is the posterior probability of the i^{th} unigram/bigram p_i, given language L_j. Returning to equation (1), $E(\cdot)$ is the prior language grouping information of language L_a and L_b according to the Ethnologue linguistic community [25] and is given by:

$$E(L_a, L_b) = \begin{cases} 1, & L_a \text{ and } L_b \in G \\ 0.5, & otherwise \end{cases} \qquad (3)$$

Here, G is a language group defined in Ethnologue. These constant prior probability values were selected empirically [17]. Equation (1) is iteratively used to compute the $k \times k$ square similarity matrix $S_{p,q}$, where p and q represents level and node respectively and N_s is the total number of languages in a language cluster. Following this, the agglomerative clustering algorithm was used to find the language clusters and develop subsequent levels. In Figure 2, the languages in grey blocks with solid lines represent the initial hierarchical structure.

3.1. Accommodating new Language in Hierarchy

The initial hierarchical structure of 28 languages was incrementally expanded in 7 steps to include 14 new target languages, which were randomly selected from NIST 2007, 2009, 2011 and 2015 database. New languages, L_n, were accommodated in the existing hierarchical structure by first using equation (1) to find their similarities with other languages in each language cluster, and the similarity matrix ($S_{p,q}$) was updated accordingly. Following this, an agglomerative clustering algorithm [14, 17] was used again to determine the language group of new languages L_n. The criterion for including a new language in the language group is described in equation (4).

$$S_{x,y \in k}(L_x, L_y) - S_{z \in k}(L_z, L_n) < \beta \qquad (4)$$

Here, k is total number of languages in a language cluster and β is a fixed threshold selected empirically for our work as 0.05. As an example, Figure 2 highlights the inclusion of two of the 14 new languages in the hierarchical structure, Arabic Egyptian (AE) and Hindi (HI). In the first level, languages AE and HI are included in language group G1 and G9 respectively. Following this, languages AE and HI are grouped with Arabic Levantine (AL) and Urdu (UR) respectively as they show maximum similarity. In Figure 2, languages in coloured blocks represent the newly added target languages.

3.2. Training Hierarchical Nodes

This section describes the novel approach used to independently train the hierarchical nodes by computing the language cluster specific features tuned at each node. The previous approaches towards HLID make use of shared features at each node extracted from one TV space and Universal Background Model (UBM), which require all nodes to be updated on scaling to new languages [16, 17]. However, in the proposed training approach nodes are only updated upon inclusion of new languages under that node.

In a HLID system, the LID task is divided into several subtasks to be carried out at each node, involving distinguishing between hypothesis languages belonging to each specific node. Classifiers trained at each nodes aim to model differences between languages/language groups associated with a specific node. For instance, $(\text{node})_{11}$ seen in Figure 2 models the differences between broad language groups (denoted as G1 to G9) while the subsequent $(\text{node})_{24}$ models the differences between sub-clusters of languages within G5 (denoted as C_9 and C_{10}). Finally, $(\text{node})_{3,11}$ in the last level models the differences between languages SE and SA which together constitute language cluster C_{10}. The proposed training approach introduces the language cluster specific features tuned at a node to capture the differences between underlying languages/language groups associated to each node. These feature sets are extracted by transforming i-vectors using DNNs trained at each node as shown in Figure 3. The DNN framework consists of 2 hidden layers of 200 dimensions followed by a softmax layer, which are trained using languages/language groups at that node as training targets. The initial hierarchical structure is trained using this approach. Later, new languages are accommodated in a hierarchical structure by either training a new DNN to extract language cluster-specific features at a new node or by upgrading an existing DNN. For example, Figure 2 shows that two DNN are trained at hierarchy $(\text{node})_{33}$ and $(\text{node})_{3,18}$ to include the languages AE and HI respectively. It may not be critical to upgrade the DNN at hierarchy $(\text{node})_{21}$, $(\text{node})_{28}$ and $(\text{node})_{11}$ as existing language in language cluster G1 and G9 might capture the enough variability to make discrimination with other language groups. Nevertheless, both training possibilities are explored in this work.

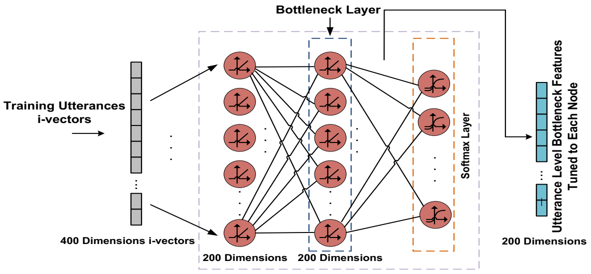

Figure 3: *Language Cluster Specific Features Using DNN*

4. Experimental Setup

4.1. Database

In this work, experiments were conducted to investigate the scalability of LID systems, requiring a comparatively large number of languages. For this purpose, 42 languages were selected from NIST 2007, 2009, 2011 and 2015 LRE datasets [5] as listed in Table 1. As previously mentioned, the initial LID was developed on 28 languages randomly selected from the 42 languages. Following this, 14 remaining languages were included in 7 steps. For development purposes, 10 conversations from each language were randomly chosen. The test set consisted of mixed duration utterances of 30, 10 and 3 seconds, as provided by NIST for previous LRE evaluations.

Table 1: *Target Languages*

American English	Amharic	Arabic Egyptian	Arabic Iraqi	Arabic Levantine	Arabic Maghrebi
Arabic Modern Standard	Bengali	Bosnian	Brazilian Portuguese	British English	Cantonese
Caribbean Spanish	Croatian	Czech	Dari	European Spanish	Farsi
German	Haitian Creole	Hausa	Hindi	Indian English	Japanese
Korean	Lao	Latin American Spanish	Mandarin	Min	Pashto
Polish	Punjabi	Russian	Tamil	Thai	Turkish
Ukrainian	Urdu	Vietnamese	West African French	Wu	Georgian

4.2. Feature Extraction

In this work, we have utilized four i-vector systems based on four different front-ends: Hungarian (HU) and Czech (CZ) PLLR features, MFCCs and lastly bottleneck features. The PLLR and MFCC cases were augmented with SDCs and estimated as outlined in [14]. The frame level BNFs of 42 dimensions were extracted using a DNN implemented with the Kaldi toolkit [26]. The DNN was trained on 300 hours of Switchboard-I data and used 13 dimensional MFCC's features as input [10]. The DNN was comprised of 5 layers with 1024 nodes in each layer, except the 4th layer which served as a 42 node bottleneck layer. A Hungarian TRAPs/NN phone decoder [27] was used to obtain phonetic transcription training data and bigram and trigram dictionaries were constructed using the SRILM lattice tool [28]. A reduced set of unigrams and bigrams of sizes 47 and 3947 respectively were chosen by discarding phoneme sequences that did not occur in the training data from any of the languages. All i-vectors based on these front-ends were 400 dimensional as in [17] and estimated using UBMs with 1024 mixture components.

4.3. Classification

The hierarchical framework allows for different front-ends to be employed at different nodes for classification. In the system presented in this paper, at each node the front-end is chosen as one of or a combination of the four features sets described in Section 4.2 based on J-Measures [29] of these features estimated from a development dataset. The selected individual or combination of features were transformed into 200-dimensional utterance level bottleneck features using the DNN described in Section 3.2. In this work, the choice of back-end in the hierarchical structure was restricted to GPLDA [14] and SVM [15] (one vs. all approach using Radial Basis Kernel) so as to make the comparison fair with single-level LID systems. Both back-ends were employed separately in two sets of HLID experiments. Both back-ends also operate on utterance level BNFs estimated from the chosen front-end as described in Section 3.2. At each node of the hierarchical framework, the log likelihoods (LLs) of all languages/language groups that were considered in that node were estimated as in [16]. Multiclass calibration models were then estimated from these LLs, using the FoCal toolkit, on a development data set and applied to the test set. Finally these calibrated LLs were converted into log likelihood ratios (LLRs) as defined by NIST [4]. The average cost performance (C_{avg}) and log likelihood ratio cost (C_{llr}), as defined by NIST [4] were employed as performance metrics in this paper.

4.4. Baseline System

The performance of the proposed HLID framework was compared with a baseline system, developed using the previous approach described in [14, 15], and comprised of four sub-systems fused at the score level. The four sub-systems were all i-vector-GPLDA systems that each used a different front-end. The four front-ends were: two acoustic (MFCC and BNF) and two phonotactic front-ends (PLLR using Hungarian and Czech phonemes) as described in Section 4.2.

5. Results

In this work, experiments were conducted to address the scalability of LID systems using single level and hierarchical frameworks. In the baseline single-level LID framework (SLID), experiments were conducted with and without re-training of UBM and TV space upon inclusion of new target languages. Similarly, in the hierarchical framework, experiments were conducted with and without re-training of a specific hierarchy tree to include new target languages. The experiment was also conducted to compare the performance of GPLDA and SVM back-ends when increasing the number of target languages in the LID system.

Table 2: *Single level and Hierarchical LID System performance using GPLDA backend.*

Number of Languages	100* C_{avg} / C_{LLR}			
	Single Level LID System (Baseline)		Hierarchical LID System	
	without re-training	with re-training	re-training specific node	re-training hierarchy tree
28	24.2 / 0.74	24.2 / 0.74	17.1 / 0.55	17.1 / 0.55
30	25.9 / 0.76	24.9 / 0.74	17.8 / 0.56	17.5 / 0.55
32	26.8 / 0.77	25.7 / 0.75	18.1 / 0.56	17.9 / 0.56
34	28.3 / 0.79	27.1 / 0.77	18.7 / 0.58	18.3 / 0.57
36	31.1 / 0.84	28.6 / 0.78	19.2 / 0.59	18.5 / 0.57
38	32.6 / 0.87	29.8 / 0.80	19.7 / 0.60	18.6 / 0.58
40	33.8 / 0.88	31.2 / 0.82	20.3 / 0.61	19.1 / 0.59
42	35.1 / 0.90	32.3 / 0.83	20.5 / 0.61	19.3 / 0.60

Table 2 shows the performance of single-level and hierarchical LID systems developed initially on 28 languages using GPLDA backend and incrementally expanded to include 14 languages in 7 steps. The results show 29.1% and 23.5% of performance degradation on increasing the number of languages from 28 to 42 in single level framework with and without updating the UBM and TV space respectively. Whereas in the hierarchical framework, LID system performance degrades by 16.6% and 11.4% with only re-training the node and complete hierarchy tree to which new target languages belongs.

Table 3 shows the performance of SLID and HLID systems using the SVM backend one vs. all training approach. The results show 22.8% and 13.4% performance degradation after accommodating the 14 new target languages in single level framework with and without updating the UBM and TV space respectively. In the hierarchical framework, LID system performance degrades by 12.4% and 5.8% with re-training only the node and complete hierarchy tree to which new target languages belongs respectively. The results show that hierarchical framework provides 12.1% and 7.6% less performance degradation using the GPLDA and SVM backends respectively over the single level framework on scaling up the LID system from 28 to 42 target languages.

Table 3: *Single level and Hierarchical LID System performance using SVM backend*.

Number of Languages	100* C_{avg} / C_{LLR}			
	Single Level LID System (Baseline)		Hierarchical LID System	
	without re-training	with re-training	re-training specific node	re-training hierarchy tree
28	25.8 / 0.76	25.8 / 0.76	17.6 / 0.56	17.6 / 0.56
30	27.7 / 0.78	26.1 / 0.77	18.1 / 0.57	17.8 / 0.56
32	28.5 / 0.79	26.7 / 0.78	18.8 / 0.56	18.1 / 0.56
34	29.3 / 0.80	27.3 / 0.78	19.1 / 0.57	18.2 / 0.57
36	30.6 / 0.82	27.8 / 0.79	19.5 / 0.58	18.3 / 0.57
38	31.9 / 0.84	28.4 / 0.80	19.6 / 0.59	18.4 / 0.57
40	32.7 / 0.85	29.1 / 0.81	19.8 / 0.59	18.5 / 0.57
42	33.4 / 0.86	29.8 / 0.82	20.1 / 0.60	18.7 /0.58

Figure 4 shows the LID system performance deterioration by increasing the number of languages from 28 to 42 in single level and hierarchical framework using GPLDA and SVM backend with re-training SLID and hierarchy branch in HLID system. It can be seen that SVM outperforms the GPLDA by 7.7% and 3.1% in the single level and hierarchical frameworks respectively, after including the 14 new target languages.

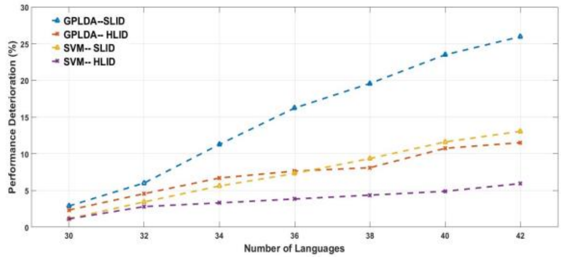

Figure 4: *Comparison of GPLDA and SVM backends.*

6. Conclusions

This paper has focused on addressing the scalability of LID system and shows that hierarchical framework is well suited for this task. This is the first investigation to show that including new target languages in hierarchical framework requires minimal re-training of LID system when compared to non-hierarchical approach. The study also indicates that single level LID system performance significantly deteriorates by increasing the number of target languages when compared to hierarchical LID system. Finally, it has been shown that SVM is best suited for large scale classification in both single level and hierarchical LID system.

7. References

[1] E. Ambikairajah, H. Li, L. Wang, B. Yin, and V. Sethu, "Language identification: A tutorial," *Circuits and Systems Magazine, IEEE,* vol. 11, no. 2, pp. 82-108, 2011.

[2] H. Li, B. Ma, and K. A. Lee, "Spoken language recognition: from fundamentals to practice," *Proceedings of the IEEE,* vol. 101, no. 5, pp. 1136-1159, 2013.

[3] X.-L. Wang, H. Zhao, and B.-l. Lu, "A meta-top-down method for large-scale hierarchical classification," *IEEE Transactions on Knowledge and Data Engineering,* vol. 26, no. 3, pp. 500-513, 2014.

[4] A. F. Martin *et al.*, "NIST Language Recognition Evaluation—Plans for 2015," in *Sixteenth Annual Conference of the International Speech Communication Association*, 2015.

[5] A. F. Martin, C. S. Greenberg, J. M. Howard, G. R. Doddington, and J. J. Godfrey, "NIST language recognition evaluation past and future," in *Odyssey: the Speaker and Language Recognition Workshop*, 2014.

[6] D. Garcia-Romero and C. Y. Espy-Wilson, "Analysis of i-vector Length Normalization in Speaker Recognition Systems," in *Interspeech*, 2011, pp. 249-252.

[7] M. Diez, A. Varona, M. Penagarikano, L. J. Rodríguez-Fuentes, and G. Bordel, "Dimensionality reduction of phone log-likelihood ratio features for spoken language recognition," in *INTERSPEECH*, 2013, pp. 64-68.

[8] M. Díez, A. Varona, M. Peñagarikano, L. J. Rodríguez-Fuentes, and G. Bordel, "On the use of phone log-likelihood ratios as features in spoken language recognition," in *SLT*, 2012, pp. 274-279.

[9] L. D'Haro, R. Cordoba, C. Salamea, and J. Echeverry, "Extended phone log-likelihood ratio features and acoustic-based i-vectors for language recognition," in *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*, 2014, pp. 5342-5346: IEEE.

[10] F. Richardson, D. Reynolds, and N. Dehak, "A Unified Deep Neural Network for Speaker and Language Recognition," *arXiv preprint arXiv:1504.00923,* 2015.

[11] R. Zazo, A. Lozano-Diez, J. Gonzalez-Dominguez, D. T. Toledano, and J. Gonzalez-Rodriguez, "Language identification in short utterances using long short-term memory (LSTM) recurrent neural networks," *PloS one,* vol. 11, no. 1, p. e0146917, 2016.

[12] W. Geng, W. Wang, Y. Zhao, X. Cai, and B. Xu, "End-to-End Language Identification Using Attention-Based Recurrent Neural Networks}," *Interspeech 2016},* pp. 2944-2948, 2016.

[13] 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: Citeseer.

[14] S. Irtza, V. Sethu, P. N. Le, E. Ambikairajah, and H. Li, "Phonemes Frequency Based PLLR Dimensionality Reduction for Language Recognition," *in Sixteenth Annual Conference of INTERSPEECH, 2015.*

[15] W. M. Campbell, E. Singer, P. A. Torres-Carrasquillo, and D. A. Reynolds, "Language recognition with support vector machines," in *ODYSSEY04-The Speaker and Language Recognition Workshop*, 2004.

[16] 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)*, 2016, pp. 5820-5824.

[17] S. Irtza, V. Sethu, S. Fernando, E. Ambikairajah, and H. Li, "Out of Set Language Modelling in Hierarchical Language Identification," *INTERSPEECH 2016,* pp. 3270-3274, 2016.

[18] B. Yin, E. Ambikairajah, and F. Chen, "Hierarchical language identification based on automatic language clustering," in *INTERSPEECH*, 2007, pp. 178-181: Citeseer.

[19] B. Yin, E. Ambikairajah, and F. Chen, "Improvements on hierarchical language identification based on automatic language clustering," in *Acoustics, Speech and Signal Processing, 2008. ICASSP 2008. IEEE International Conference on*, 2008, pp. 4241-4244: IEEE.

[20] A. Bouguettaya, Q. Yu, X. Liu, X. Zhou, and A. Song, "Efficient agglomerative hierarchical clustering," *Expert Systems with Applications,* vol. 42, no. 5, pp. 2785-2797, 2015.

[21] P. Matejka *et al.*, "BUT system description for NIST LRE 2007," in *Proc. 2007 NIST Language Recognition Evaluation Workshop*, 2007, pp. 1-5.

[22] E. Singer *et al.*, "The MITLL NIST LRE 2011 language recognition system," in *Odyssey*, 2012, pp. 209-215.

[23] R. W. Ng *et al.*, "The Sheffield language recognition system in NIST LRE 2015," in *Proceedings of The Speaker and Language Recognition Workshop Odyssey 2016*, 2016, pp. 181-187: ISCA.

[24] S.-H. Cha, "Comprehensive survey on distance/similarity measures between probability density functions," *City,* vol. 1, no. 2, p. 1, 2007.

[25] M. P. Lewis, G. F. Simons, and C. D. Fennig, *Ethnologue: Languages of the world.* SIL international Dallas, TX, 2009.

[26] D. Povey *et al.*, "The Kaldi speech recognition toolkit," in *IEEE 2011 workshop on automatic speech recognition and understanding*, 2011, no. EPFL-CONF-192584: IEEE Signal Processing Society.

[27] P. Schwarz, "Phoneme recognition based on long temporal context," ed: Faculty of Information Technology BUT, 2009.

[28] A. Stolcke, "SRILM-an extensible language modeling toolkit," in *Interspeech*, 2002, vol. 2002, p. 2002.

[29] S. Nicholson, B. P. Milner, and S. J. Cox, "Evaluating feature set performance using the f-ratio and j-measures," in *EUROSPEECH*, 1997, vol. 97, pp. 413-416.