END-TO-END 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

Recently, hierarchical language identification systems have shown significant improvement over single level systems in both closed and open set language identification tasks. However, developing such a system requires the features and classifier selection at each node in the hierarchical structure to be hand crafted. Motivated by the superior ability of end-to-end deep neural network architecture to jointly optimize the feature extraction and classification process, we propose a novel approach developing an end-to-end hierarchical language identification system. The proposed approach also demonstrates the in-built ability of the end-to-end hierarchical structure training that enables an out-of-set language model, without using any additional out-of-set language training data. Experiments are conducted on the NIST LRE 2015 data set. The overall results show relative improvements of 18.6% and 27.3% in terms of C_{avg} in closed and open set tasks over the corresponding baseline systems.

Index Terms— End-to-end framework, Hierarchical framework, Out-of-set language modelling, Language identification

1. INTRODUCTION

The task of automatic language recognition is to identify or verify the language being spoken from a group of possible languages [1]. The most widely adopted approaches to language identification (LID) use acoustic and phonotactic information [1, 2]. Specifically, current systems employ the i-vector framework trained on both acoustic and phonotactic front-ends. Mel frequency cepstral coefficients (MFCCs) and phone log likelihood ratios (PLLRs) continue to be the most commonly used of these types of front-ends and recently bottleneck features (BNF) have exhibited promising performance [2-4]. These features represent the short term spectral or phonetic information of a speech signal. Longer term information is then typically captured through the use of supervector representations of utterances or through total variability factor analysis in the i-vector framework [5, 6].

More recently, deep neural networks (DNNs) have been employed either in the front-end using BNF or in end-to-end architectures for language identification [4, 7]. Deep learning approaches e.g. convolutional neural network (CNN), recurrent neural network (RNN) and long short-term memory (LSTM) have also shown competitive performance [8, 9, 10]. These deep learning approaches have become popular among researchers and have been successfully used to develop end-to-end LID systems, which jointly optimize the feature extraction and system backend [8].

The end-to-end LID system consists of various combinations of CNN, RNN variants (e.g. vanilla RNN, LSTM and bi-directional LSTM (BLSTM)) or fully connected DNN layers including softmax layer [8, 11]. Most commonly in these systems, a raw speech signal or spectrogram is first processed by CNN layers to extract robust low level features. [11]. The CNN consists of a set of shared weights applied along the entire input space, which processes a portion of the input signal, followed by the max pooling layer, which generates a lower resolution version of convolutional filter outputs by computing the maximum value of filter activations within a specified window. The CNN layers are cascaded with one of the RNN variants to model long-term temporal information [8, 9, 11-13]. In terms of overall system performance, LSTM and BLSTM outperforms the other RNN variants. The final stage of an end-to-end system consists of fully connected DNN and softmax layers to extract hierarchical representations that benefit discrimination between classes. The softmax layer maps the hidden states into interpretable probability vector of target languages. These systems are all single level approaches where all language hypotheses are treated in parallel [2, 14]. As opposed to single level system where each language model is built independently, hierarchical structure incorporates prior knowledge about language family, and allows for sharing of training data across languages within the same cluster/group [11].

The hierarchical LID (HLID) framework has been previously proposed as an alternative approach that makes use of language similarity information to identify languages through a multi-level decision [14-16]. In this way, the LID problem is solved through a top-down hierarchy of smaller sub-problems, with initial high-level decisions pertaining to identification of language groups followed by identification of specific languages [14]. Despite the superior performance of the hierarchical framework, it requires significant extra effort to determine appropriate features and classifiers for the languages/groups at each node. One possible solution is to jointly optimize the feature extraction and classifier at

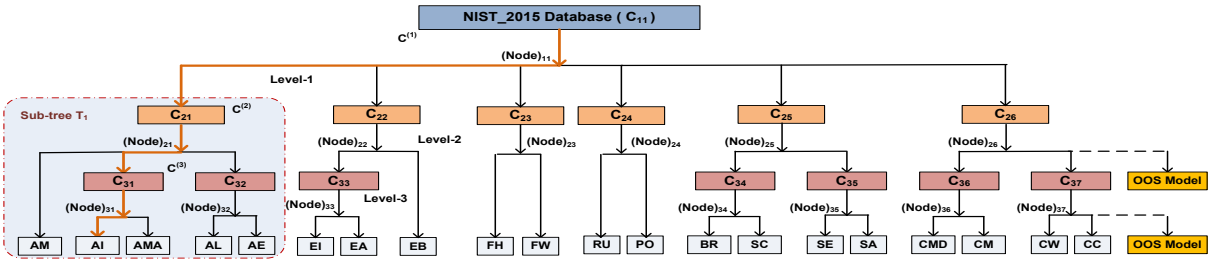

Figure 1: *Hierarchical language identification framework on 2015 database.*

each node to automate the training of hierarchical LID system. The question arises as how to develop and optimize a DNN for such a system.

In this paper, we propose two novel approaches to training such end-to-end HLID systems by jointly optimizing the nodes that are under same sub-tree in the hierarchical structure. The first approach combines the loss function associated with each node in the same sub-tree. The second approach combines the language/group posteriors in the prediction layer to form a unique objective function for each sub-tree. This work also explores the effectiveness of this structure in recognizing out-of-set (OOS) languages without using any additional non-target, or OOS, language data.

2. HIERARCHICAL FRAMEWORK

In this work, a previously developed hierarchical clustering algorithm is used again to form the initial hierarchical structure [17]. The algorithm uses pairwise similarities between languages/groups using phonotactic and linguistic information. The similarity $S(\cdot)$ between a language pair (ℓ_a, ℓ_b) is computed as

$$S(\ell_a, \ell_b) = \left(1 - K_s(\ell_a, \ell_b)\right) \times E(\ell_a, \ell_b) \quad (1)$$

where integers $a, b \in [1, L]$, L is the total number of languages, and $K_s(\cdot)$ is the symmetric K-divergence between the phoneme probability distribution of ℓ_a and ℓ_b. $E(\cdot)$ is the prior language grouping information of language ℓ_a and ℓ_b according to the Ethnologue linguistic community [25] and is given by:

$$E(\ell_a, \ell_b) = \begin{cases} 1, & \ell_a \text{ and } \ell_b \in C \\ 0.5, & otherwise \end{cases} \quad (2)$$

Here, C is a language group as defined in Ethnologue. These constant prior probability values were selected empirically, as in previous work [14, 15]. Figure 1 shows the hierarchical structure developed on the NIST LRE 2015 database.

It can be seen from Figure 1 that the LID task is divided into several subtasks which are carried out at each node. At each specific node, we distinguish between hypothesis languages/groups belonging to the node. Referring to Figure 1, features tuned at $(\text{node})_{11}$ model the differences between broad language groups (denoted C_{21} to C_{26}) while the subsequent $(\text{node})_{25}$ models the differences between sub-clusters of languages within C_{25} (denoted C_{34} and C_{35}). Finally, features in $(\text{node})_{34}$ in the last level model the differences between languages Brazilian (BR) and Spanish Caribbean (SC). Language cluster specific features tuned at each node have been shown to capture the differences between underlying languages/groups effectively [18]. However, the existing approach requires the development of a language cluster feature extractor at each node in the hierarchical structure. In this way, the selection of an appropriate classifier at each node puts an extra burden on the development of the HLID system. This work presents an end-to-end approach to develop a LID system which jointly optimizes the language cluster specific features and classifiers at each node in hierarchical structure.

3. PROPOSED END-TO-END HIERARCHICAL NETWORK

Figure 2 shows the proposed end-to-end architecture of a HLID system. It consists of feature extraction, and language group specific networks. The feature extraction network consists of two CNN layers to extract robust features from the spectrogram and is shared between all the nodes in hierarchical structure. The language group specific network consists of one LSTM and three fully connected DNN layers. The motivation of the language group specific network is to capture the language's/group's specific long-term temporal information at each node as shown in Figure 1. Finally, the softmax layer is trained using languages/groups at that node as training targets. We propose two novel approaches to jointly optimizing all nodes that are under the same sub-tree in the hierarchical structure by either combining respective prediction loss (Section 3.1) or combining the posteriors of languages/groups on the path from the root node of the hierarchical tree to the leaf nodes (Section 3.2).

3.1. Approach-I: Optimizing Combined Prediction Loss

This approach aims to combine the prediction loss of each language group's specific network to define a loss function for each sub-tree in the hierarchy. This approach leads each sub-tree to have its own loss function to be optimized. As an example from Figure 1, the loss function associated with sub-tree T_1 is the summation of the individual prediction losses of language group specific networks at four nodes, namely $(\text{node})_{11}$, $(\text{node})_{21}$, $(\text{node})_{31}$ and $(\text{node})_{32}$.

The language group specific network at $(\text{node})_k$ produces a prediction loss for the i^{th} speech example $(x_i, y_{i,k})$ as follows:

$$\mathcal{L}_k^i(\theta_f, \theta_k) = -\log\left(\left(G_k\left(G_f(x_i;\, \theta_f);\theta_k\right), y_{i,k}\right)\right) \quad (3)$$

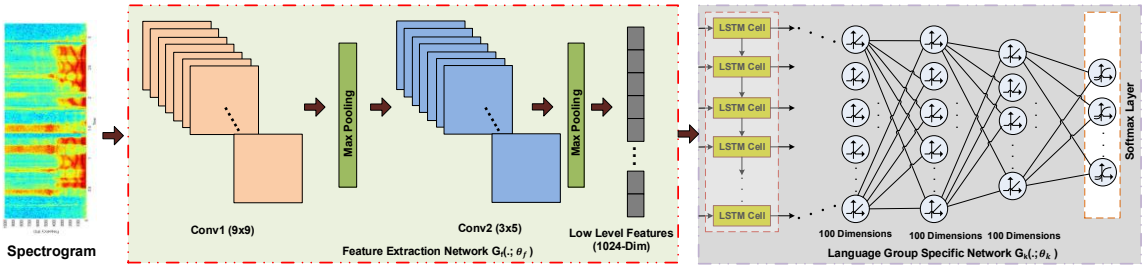

Figure 2: *End-to-end hierarchical language identification network*

where $G_f(.;\theta_f)$ represents the feature extraction network that learns a function to map a spectrogram into low dimensional feature vector at node *k,* which maps a given feature vector into a probability that assigns it to the target feature representation, and $G_k(.;\theta_k)$ represents the language/group labels $y_{i,k}$ associated with node *k*. Finally, θ_f and θ_k are feature extractor and k^{th} language group specific network parameters. Here *'k'* represents the node number assigned to each node in Figure 1 e.g. k_{11}, k_{21} and so on.

Training the neural network then leads to the parallel optimization of the objective function for each sub-tree in the hierarchy structure:

$$E_{T_j}\left(\theta_f,\theta_{T_j}\right)=\min_{\theta_f,\theta_{T_j}}\frac{1}{I}\sum_{i=1}^{I}\mathcal{L}_{T_j}^{i}\left(\theta_f,\theta_{T_j}\right) \tag{4}$$

Here, $\mathcal{L}_{T_j}^{i}$ is the loss function of j^{th} sub-tree, defined as:

$$\mathcal{L}_{T_j}^{i}\left(\theta_f,\theta_{T_j}\right)=\sum_{k=1}^{K}\mathcal{L}_{k}^{i}(\theta_f,\theta_k) \tag{5}$$

E_{T_j} and θ_{T_j} are the objective functions and set of network parameters for the j^{th} sub-tree in the hierarchical structure, *I* is the total number of speech examples and *K* is set of all nodes in the j^{th} sub-tree.

3.2. Approach-II: Optimizing Combined Objective Function

In contrast to Section 3.1, where we iterate over all nodes *k* in a sub-tree T_j, here we aim to iterate over nodes/clusters in a particular path from root to leaf of the sub-tree. We introduce the superscript *(n)* to denote the number of levels *n* in the sub-tree. An example of such a path is highlighted in red in Figure 1 for the Arabic Iraqi (AI) language.

This approach merges the prediction (softmax) layers of all nodes in a path from root cluster to leaf language, to compute the language posteriors and form a single objective function to be optimized. In a HLID system, the language posterior of each target language is computed as the chain product of the conditional probabilities of a target language ℓ_t or cluster/group $C^{(n)}$, given the parent $C^{(n-1)}$, on the path from root to leaf. The posterior of $\ell_t \in T_j$ for a given test x_i is

$$P(\ell_t|x_i)=P\left(\ell_t|C^{(N)},x_i\right)\left(\prod_{n=2}^{N}P\left(C^{(n)}|C^{(n-1)},x_i\right)\right) \tag{6}$$

where each term of the product $P\left(C^{(n)}|C^{(n-1)},x_i\right)$ is computed as:

$$P\left(C^{(n)}|C^{(n-1)},x_i\right)=\left(G^{(n-1)}\left(G_f\left(x_i;\,\theta_f\right);\theta^{(n-1)}\right),x_i\right) \tag{7}$$

$P\left(\ell_t|C^{(N)},x_i\right)$ is the conditional probability of the leaf language given its parent cluster. The posterior probability of AI (path highlighted red in Figure 1), given an input utterance x_i, is expanded using node subscript notation as:

$$P(AI|x_i)=P(AI|C_{31},x_i)P(C_{31}|C_{21},x_i)P(C_{21}|C_{11},x_i) \tag{8}$$

The posteriors for all languages in sub-tree T_j are then concatenated to form a posterior vector

$$P(\boldsymbol{\ell}|x_i)=[P(\ell_1|x_i)\;\;P(\ell_2|x_i)\,..P(\ell_T|x_i)]^T \tag{9}$$

Finally, by training the hierarchical structure, we optimize the following objective function for each sub-tree:

$$E_{T_j}(\theta_f,\theta_{T_j})=\min_{\theta_f,\theta_{T_j}}\frac{1}{I}\sum_{i=1}^{I}\mathcal{L}_{T_j}^{i}(P(\boldsymbol{\ell}|x_i);\theta_f,\theta_{T_j}) \tag{10}$$

where $\mathcal{L}_{T_j}^{i}$ is the overall loss function of sub-tree T_j, and $P(\boldsymbol{\ell}|x_i)$ is used in the overall loss function, taking the place of the log argument to equation (3), to compute the overall prediction loss ($\mathcal{L}_{T_j}$) for each sub-tree (T_j).

4. OOS LANGUAGE MODELLING IN END-TO-END NETWORK

In this section, we propose a way to model OOS languages at multiple levels of the hierarchy where the OOS model at each node is a model of all languages not considered at that node. It has been shown that the OOS models at each node provide a different background model that is specific to the languages considered at each node. Consequently, the hierarchical structure can detect OOS languages more reliably overall [17]. In the proposed end-to-end network, OOS language models are incorporated into the nodes in the second and third level of the hierarchical structure as shown in Figure 1.

The OOS models are developed at each node by using the same training approach as described in Section 3. As the end-to-end network is jointly optimized, language group specific networks at each node process all training batches even if they do not contain data from the languages/groups associated with a specific node. Therefore, all the softmax layers in the second and third levels of the hierarchical structure are developed using L_k+1 language models, where L_k is the number of target languages in the k^{th} node and the one for the OOS model. These language group

specific OOS models are trained on the data from languages that are not present in that group e.g. OOS model for C_{26} is developed on training data for C_{21} to C_{25}.

5. EXPERIMENTAL SETUP

The LID experiments reported in this work were carried out on the NIST LRE 2015 dataset [19]. The closed set results are reported on the NIST LRE 2015 dataset as per the fixed test conditions given in [19] that involve 20 target languages. The proposed systems are evaluated in terms of C_{avg} and C_{LLR} as per LRE protocol (lower these values correspond to better system performance). 10 conversations from each language were randomly chosen for development purposes. The open set LID experiments require additional test data corresponding to OOS languages [20]. This additional OOS test data, of 30, 10 and 3 second durations from 17 different languages, was selected from the NIST LRE 2007 and 2011 datasets as in previous study [14]. They include the following languages, Bengali, Czech, Dari, Farsi, Thai, Urdu, Japanese, Vietnamese, Ukrainian, Hindi, Punjabi, Pashto, Tamil, Turkish, German, Korean and Lao.

5.1. Configuration of End-to-End Network

This section describes the configuration of the CNN, LSTM and fully connected layers to form an end-to-end LID system. Input to this network is a spectrogram of speech recordings consisting of 128 frequency bins for each 30ms frame, with a 50% overlapping Hamming window. These speech spectrograms are fed into 2D convolutional layers and 2D max pooling layers with a filter size of 9x9 and 3x5 respectively, to extract robust low-level features. These features are shared between all language group specific networks, consisting of one LSTM layer of 256 memory blocks, and a four layer DNN comprising of two hidden layers of 100 dimension, one bottleneck hidden layer of 42 dimensions and a softmax layer, to predict languages/groups at each node. The likelihood of each language is computed by averaging the frame level likelihood for each test utterance. Each hidden layer uses the rectified linear unit (ReLU) activation function. A momentum optimizer [7] is used to optimize the network using the dropout regularization method. The design and selection of end-to-end network parameters is consistent with the previously developed single level end-to-end system in [8].

5.2. Baseline System

The performance of the end-to-end HLID system is compared to an equivalent single level end-to-end system. This baseline system also uses the same end-to-end architecture trained to predict all the target languages. For the open set experiments, the OOS language model in the baseline system was estimated by using the additional diverse OOS languages training data as described in [17].

6. EXPERIMENTS

Two sets of experiments were conducted: 1) closed set language detection, and 2) open set language detection. The closed set experiments were conducted to a) quantify the performance of the end-to-end hierarchical structure without OOS languages, and b) investigate the effectiveness of the proposed optimization methods in Section 3. The open set experiments were conducted to investigate the proposed OOS modelling approach in an end-to-end HLID system. We follow the NIST LRE evaluation protocol [19] when reporting the results.

6.1. Closed Set Detection Results

Table 1 compares the performance between the end-to-end baseline and the proposed HLID systems in the closed set experiments. The HLID systems using approaches I and II outperform the baseline system by 14.7% and 18.6% respectively in terms of C_{avg}.

Table 1: *Closed set detection results on NIST LRE 2015.*

Language Groups	100* C_{avg} / C_{LLR}		
	Single Level (Baseline)	Hierarchical (Approach-I)	Hierarchical (Approach-II)
Arabic	21.4 / 0.67	19.4 / 0.62	18.8 / 0.60
Chinese	19.1 / 0.61	13.9 / 0.44	12.7 / 0.42
English	12.5 / 0.42	9.8 / 0.38	8.5 / 0.35
French	41.2 / 0.94	37.4 / 0.90	36.9 / 0.89
Slavic	5.7 / 0.17	4.2 / 0.15	3.7 / 0.14
Iberian	22.9 / 0.69	20.1 / 0.63	19.2 / 0.61
Overall	20.4 / 0.58	17.4 / 0.52	16.6 / 0.50

6.2. Open Set Detection Results

Table 2 compares the performance between the end-to-end baseline and the proposed HLID systems in the open set experiments. We observe that the inclusion of OOS leads to performance degradation across all settings when comparing with the results in Table 1. The results show that the proposed hierarchical OOS modeling outperforms the explicit OOS modeling in the end-to-end system.

Table 2: *Open set detection results on NIST LRE 2015.*

Language Groups	100* C_{avg} / C_{LLR}		
	Single Level (Baseline)	Hierarchical (Approach-I)	Hierarchical (Approach-II)
Arabic	26.7 / 0.74	21.1 / 0.67	20.2 / 0.66
Chinese	25.3 / 0.72	15.7 / 0.47	14.5 / 0.45
English	18.5 / 0.59	13.5 / 0.43	12.9 / 0.42
French	47.2 / 1.30	40.2 / 0.91	39.6 / 0.91
Slavic	10.9 / 0.40	6.1 / 0.28	4.9 / 0.25
Iberian	27.1 / 0.75	23.2 / 0.71	21.4 / 0.67
Overall	25.9 / 0.75	19.9 / 0.57	18.9 / 0.56

7. CONCLUSION

This paper has focused on automating the language cluster specific feature extraction and classifier selection process at each node in a hierarchical language identification system. The study shows that the proposed approaches better optimize the hierarchical structure. The results also indicate that the development of out-of-set language models in a hierarchical framework is better able to reject unknown languages than a non-hierarchical approach.

8. REFERENCES

[1] E. Ambikairajah, H. Li, L. Wang, B. Yin, and V. Sethu, "Language identification: A tutorial," *Circuits and Systems Magazine, IEEE,* vol. 11, 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, pp. 1136-1159, 2013.

[3] 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.

[4] F. Richardson, D. Reynolds, and N. Dehak, "A unified deep neural network for speaker and language recognition," *arXiv preprint arXiv:1504.00923,* 2015.

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

[6] 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.

[7] 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.

[8] T. N. Trong, V. Hautamäki, and K. A. Lee, "Deep language: a comprehensive deep learning approach to end-to-end language recognition," in *Odyssey: the Speaker and Language Recognition Workshop*, 2016.

[9] 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, p. e0146917, 2016.

[10] Y. Zhang, M. Pezeshki, P. Brakel, S. Zhang, C. L. Y. Bengio, and A. Courville, "Towards end-to-end speech recognition with deep convolutional neural networks," *arXiv preprint arXiv:1701.02720,* 2017.

[11] M. Jin, Y. Song, and I. V. McLoughlin, "End-to-end DNN-CNN Classification for Language Identification," in *Proceedings of the World Congress on Engineering*, 2017.

[12] J. Pešán, L. Burget, and J. Černocký, "Sequence Summarizing Neural Networks for Spoken Language Recognition," *Interspeech 2016,* pp. 3285-3288, 2016.

[13] J. Gonzalez-Dominguez, I. Lopez-Moreno, H. Sak, J. Gonzalez-Rodriguez, and P. J. Moreno, "Automatic language identification using long short-term memory recurrent neural networks," in *Fifteenth Annual Conference of the International Speech Communication Association*, 2014.

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

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

[16] 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.

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

[18] S. Irtza, V. Sethu, E. Ambikairajah, and H. Li, "Investigating Scalability in Hierarchical Language Identification System," *Proc. Interspeech 2017,* pp. 2581-2585, 2017.

[19] A. F. Martin, C. S. Greenberg, J. M. Howard, D. Bansé, G. R. Doddington, J. Hernández-Cordero, *et al.*, "NIST Language Recognition Evaluation—Plans for 2015," in *Sixteenth Annual Conference of the International Speech Communication Association*, 2015.

[20] Q. Zhang and J. H. Hansen, "Training candidate selection for effective rejection in open-set language identification," in *Spoken Language Technology Workshop (SLT), 2014 IEEE*, 2014, pp. 384-389.