Interspeech 2018
2-6 September 2018, Hyderabad

Feature Representation of Short Utterances based on Knowledge Distillation for Spoken Language Identification

Peng Shen, Xugang Lu, Sheng Li, Hisashi Kawai

Institute of Information and Communications Technology, Japan

peng.shen@nict.go.jp

Abstract

The performance of spoken language identification (LID) on short utterances is drastically degraded even though model is completely trained on short utterance data set. The degradation is because of the large pattern confusion caused by the large variation of feature representation on short utterances. In this paper, we propose a teacher-student network learning algorithm to explore discriminative features for short utterances. With the teacher-student network learning, the feature representation for short utterances (explored by the student network) are normalized to their representations corresponding to long utterances (provided by the teacher network). With this learning algorithm, the feature representation on short utterances is supposed to reduce pattern confusion. Experiments on a 10-language LID task were carried out to test the algorithm. Our results showed the proposed algorithm significantly improved the performance.
Index Terms: Knowledge distillation, transfer learning, feature representation, short utterances, spoken language identification.

1. Introduction

Spoken language identification (LID) is a task to determine which language is being spoken within a speech utterance [1, 2]. LID typically acts as a pre-processing stage for a wide range of multilingual speech processing systems, such as spoken language translation and multilingual speech recognition. In most of these applications, LID is used in real-time scenarios so computational cost is often critical. Therefore, improving the performance of LID on short utterances is one of the important tasks.

LID techniques have been widely investigated and progressed recently. I-vector techniques with conventional classifiers, such as, support vector machine (SVM), probabilistic linear discriminant analysis (PLDA), and deep neural network (DNN), have demonstrated their effectiveness and obtained state-of-the-art performance in many systems, especially on relative longer utterance tasks [3, 4, 5, 6, 7, 8, 9, 10]. However, on short utterance LID tasks, the performance of the i-vector-based approaches often degrade dramatically. Recently, end-to-end approaches with convolutional neural networks (CNN), recurrent neural networks (RNN), and attention-based neural networks have been investigated on LID tasks[11, 12, 13, 14]. For short utterance LID tasks, the end-to-end approaches have demonstrated more impressive performance than i-vector-based approaches [12, 13]. For example, Fernando et al. proposed bidirectional long short term memory network (biLSTM) for short durations (3 seconds) LID tasks by modelling temporal dependencies between past and future frame based features in short utterances [13]. Lozano-diez et al. used deep convolutional neural networks (DCNN) for short test durations (segments up to 3 seconds of speech) [12]. Compared with long utterances, the feature representation of short utterances has large variation, that prevents the model from generalizing well. How to improve the generalization of the model on short utterances is still a challenge task.

In this work, we focus on short utterance (from 0.5s to 2.0s) LID tasks using DCNN-based end-to-end approach similarly to [12]. Generally, DCNN models include several convolution layers connected to one or several fully connected (FC) layers. The convolution layers can be considered as feature extraction layers, and FC layers as classification layers. Different from frame by frame training approaches, DCNN models use longer fixed length utterances as inputs. The utterance level-based DCNN model can capture high level discriminative representation that is expected to be used for improving the generalization ability of the model. However, as the input sentence becomes shorter, the performance of the DCNN model decreases rapidly even the model is completely trained on short utterance dataset.

Inspired by previous works of knowledge distillation [15, 16], we proposed a knowledge distillation-based training approach by transferring the feature representation knowledge of a long utterance-based teacher model to a short utterance-based student model. Knowledge distillation was firstly proposed by Hinton et al. [15] by using a teacher's softened output as soft label for a compact/small student model training. Romero et al. [16] proposed a hidden layer-based knowledge distillation training, called FitNets, that uses one teacher's hidden layer's output for a deeper student network training. The knowledge distillation approaches have been already successfully applied on many tasks, such as speech recognition and image classification. In this work, the feature representation knowledge, corresponding to a hidden layer of a teacher model, is transferred to a student model to help the student model to capture robust discriminative information from short utterances. To the best of our knowledge, using knowledge distillation to transfer knowledge from long utterance-based teacher to short utterance-based student for end-to-end LID tasks has not yet been studied. We evaluated the proposed method on a 10-language dataset. Experiment results indicated that the proposed method is effective for the DCNN-based short utterance LID task.

The remainder of the paper is organized as follows. Sections 2 presents the basic knowledge distillation approach. The proposed method is described in Section 3. In Section 4, the results of experiments and analysis are given to evaluate the performance of the proposed method. Conclusion is given in Section 5.

2. Knowledge Distillation Approach

Knowledge distillation (KD) is a compression framework [15], which trains a compact student network using the output of a high-performance teacher network as soft label. The student network can explore not only the information provided by true labels, but also the knowledge learned by the teacher network.

Let $\mathbf{x}$ be a given input feature, and its corresponding la-

10.21437/Interspeech.2018-1519

bel $\mathbf{y}$, here called hard label, is a K-dimensional one-hot vector. K is the number of the target classes. Let $\mathbf{q} = [q_0, ..., q_i, ..., q_{K-1}]$ be an output softmax of a teacher network, called soft label, that is also a K-dimensional vector, q_i is the softmax probability of the i-th class, and $z_i(\mathbf{x})$ is the teacher's pre-softmax output activation of the i-th class, called logits. Then, we can describe the relationship of q_i and $z_i(\mathbf{x})$ as:

$$q_i = \frac{exp(z_i(\mathbf{x})/T)}{\sum_{j=1}^{K} exp(z_j(\mathbf{x})/T)}, \quad (1)$$

where T is a temperature that is normally set to 1. Since $\mathbf{q}$ might be very close to the one-hot code representation of the samples's hard label, a higher value of T is introduced to obtain a softer probability distribution over classes. As T becomes large, $\mathbf{q}$ tends to have a uniform distribution. The student network is trained to optimize the following loss function:

$$L_{KD} = \frac{1}{N} \sum_{\mathbf{x}} ((1-\lambda) L_{hard}(\mathbf{x}, \mathbf{y}) + \lambda L_{soft}(\mathbf{x}, \mathbf{q})), \quad (2)$$

where λ is the weight to make a balance between the hard and soft losses, and N represents the number of samples $\mathbf{x}$. For the classification task, cross entropy loss is used. Then, the cross entropy-based hard and soft losses for sample $\mathbf{x}$ can be described as:

$$L_{hard}(\mathbf{x}, \mathbf{y}) = -\mathbf{y}^T \log \mathbf{p}(\mathbf{x}), \quad (3)$$

$$L_{soft}(\mathbf{x}, \mathbf{q}) = -\mathbf{q}^T \log \mathbf{p}(\mathbf{x}), \quad (4)$$

where $\mathbf{p}(\mathbf{x})$ is a K-dimensional vector with output probability of the student model for classes, $\mathbf{y}^T$ and $\mathbf{q}^T$ is a transpose operation on $\mathbf{y}$ and $\mathbf{q}$. In the soft loss, the output of student network is also applied with the same temperature T, when it is compared to the teacher's softened output $\mathbf{q}$.

3. Feature Representation Knowledge Distillation

Conventional knowledge distillation methods use one single or ensemble multiple high-performance models as a teacher model, and transfer the knowledge of the teacher model to a compact student model [15, 16]. Different from the conventional knowledge distillation methods which focus on training small compact model, we focus on improving the performance of DCNN-based LID on short utterance tasks. Compared with LID on short utterances, the performance of LID on long utterances is better. Due to duration mismatch, the long utterance-based model cannot work well on short utterances directly.

In this work, we designed a feature representation knowledge distillation (FRKD) framework by transferring the feature representation knowledge from a long utterance-based teacher network to a short utterance-based student network for LID tasks. The proposed method is illustrated in Fig. 1. Same to [16], the hidden layer-based knowledge is used to guide the student model to capture robust discriminative feature from short utterances.

Mathematically, we choose a hidden layer's feature representation with parameter set $\mathbf{\Theta}_T$ of a teacher network, where $\mathbf{\Theta}_T = \{\mathbf{W}_T, \mathbf{b}_T\}$, and a hidden layer's feature representation with parameter set $\mathbf{\Theta}_S$ of a student network, and transfer the knowledge from the teacher to the student. Then, the teacher-student transfer learning can be optimized by minimizing the

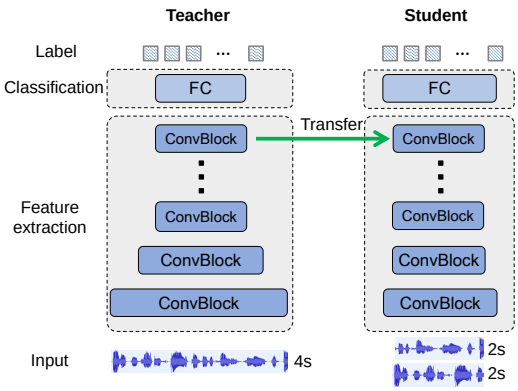

Figure 1: *The proposed feature representation knowledge distillation framework.*

following loss function:

$$L_{FRKD} = \frac{1}{N} \sum_{\mathbf{x}_S, \mathbf{x}_T} ((1-\lambda) L_{hard}(\mathbf{x}_S, \mathbf{y}) + \lambda L_{\text{kt}}(\mathbf{x}_T, \mathbf{x}_S, \mathbf{\Theta}_T, \mathbf{\Theta}_S)), \quad (5)$$

where L_{kt} is the knowledge transfer learning loss function, which can be defined as:

$$L_{\text{kt}}(\mathbf{x}_T, \mathbf{x}_S, \mathbf{\Theta}_T, \mathbf{\Theta}_S) = \|u_T(\mathbf{x}_T; \mathbf{\Theta}_T) - u_S(\mathbf{x}_S; \mathbf{\Theta}_S)\|_1, \quad (6)$$

where u_T and u_S are the teacher and student deep nested functions up to their respective selected layers with output of parameter sets $\mathbf{\Theta}_T$ and $\mathbf{\Theta}_S$, and $\|\bullet\|_1$ is the L1-norm loss. Compared with conventional knowledge distillation approaches which use the same inputs for both teacher and student models, we use $\mathbf{x}_T$ as the inputs of teacher, and $\mathbf{x}_S$ as the inputs of student, and $\mathbf{x}_S$ is short segments truncated from the long segments of the input to the teacher network, i.e., $\mathbf{x}_T$. For example, $\mathbf{x}_T$ is a four second utterance, and $\mathbf{x}_S$ is the corresponding two second utterance. In FitNets, there is a regressor layer in the student model to match the size of the teacher layer [16]. Adding the regressor to student model increases the training complexity. In this work, we focus not on building compact student model but on improving the performance of short duration utterances, therefore, we apply max-pooling to keep the selected hidden layer of the teacher and student with the same size.

The training procedure is also different from FitNets. FitNets used the transfer learning as a pre-training for the student network. In this work, Eq. 5 is a joint learning of the cross-entropy loss and the knowledge transfer loss. For the whole training procedure, firstly, we train the teacher model on long utterances by minimizing Eq. 3 with hard label. Then, the short utterance-based student model is optimized with Eq. 5.

4. Experiments

In this section, experiments were conducted to evaluate the effectiveness of the proposed method. We used a 10-language dataset of NICT to evaluate the proposed method. The spoken utterances were spoken by native speakers. We split them into training set (Train), validation set (Valid), and test set (Test). There were 100.76 hours of training data, and 24.95 hours of test data, totally. The average duration of each utterance was 7.6 seconds. The number of utterances for the training data was 45000, and for each language was 4500. For the validation and

Table 1: *Experimental data sets.*

Language	Train/Valid	Test
Burmese	4500/300 (7.27 h)	1200 (1.81 h)
Chinese	4500/300 (8.50 h)	1200 (2.13 h)
English	4500/300 (12.08 h)	1200 (3.08 h)
French	4500/300 (10.75 h)	1200 (2.67 h)
Indonesian	4500/300 (11.11 h)	1200 (2.69 h)
Japanese	4500/300 (8.92 h)	1200 (2.20 h)
Korean	4500/300 (12.23 h)	1200 (3.00 h)
Spanish	4500/300 (8.84 h)	1200 (2.14 h)
Thai	4500/300 (11.96 h)	1200 (2.96 h)
Vietnamese	4500/300 (9.10 h)	1200 (2.27 h)
ALL	45000/3000 (100.76 h)	12000 (24.95 h)

test data, it was 300 and 1200 utterances for each language. Details of the number of utterances and the data size are shown in Table 1. The utterance identification error rate (UER) was used as the evaluation criterion.

4.1. Implementation of baseline systems

We built baseline systems with conventional i-vector-based approaches and end-to-end deep learning approaches. The i-vector-based methods with support vector machine (SVM) and DNN as classifier were examined. The i-vectors were 400-dimensional vectors that obtained on the full-length duration utterances (Average 7.6s) with the script of Kaldi toolkit [17]. For SVM classifier, we used the radial basis function (RBF) kernel and a grid search with cross-validation following the work [18]. The DNN model were with two hidden layers with 512 neurons for each, and a dropout of 0.3 was applied. The mini-batch size was set to 128, and stochastic gradient descent (SGD) with learning rate 0.001 was used in this experiment.

We also compared end-to-end approaches, i.e., RNN, bidirectional RNN (biRNN) with the gated recurrent unit (GRU), and a DCNN model on four second utterances. Finally, the DCNN model was used to test on all target duration utterances, i.e., 2.0s, 1.5s, 1.0s and 0.5s. To extract the target duration utterances, power energy-based VAD was used to detect the speech, then certain duration utterances were cut with a shift same to the target duration. Then, 60-dimensional mel-filterbank features were extracted for all the prepared utterances. Finally, mean and variance normalization was applied on each utterance. For the testing data set, only the start of certain duration was cut based on the VAD results. For the RNN models, we tested different configurations (RNN and biRNN with one or two hidden layers with 256 neurons for each) and dropout with 0.0, 0.3 and 0.5. The DCNN model used for four second utterance is illustrated in Table 2. For different duration utterances, we changed the stride of max-pooling to make the last convolution layer of all the models with the same size. For the RNN and DCNN models, the mini-batch size was set to 32, RMSProp optimizer with learning rate 0.001 for model optimization. The maximum learning epoch was set to 100, and the optimal model was selected using the validation data set.

4.2. Implementation of the proposed method

To evaluate the proposed method, the same DCNN models as described in Subsection 4.1 were used. The DCNN network included seven convolution blocks and two FC blocks. The convolution layers could be considered as feature extraction layers and the FC layers as classification layers. The proposed method transferred the knowledge of feature representation (the Flatten layer) of the teacher model to the student model. The network was optimized with Eq. 5. For Eq. 6, we compared L1-norm and L2-norm-based distance metric. λ was also compared with the value of 0.1, 0.3, 0.5 and 0.7. The networks were trained using RMSProp with learning rate 0.001. The mini-batch size was set to 32. The optimal model was selected using the validation data set with maximum epoch 100.

Table 2: *The DCNN networks used in this work for four seconds input utterance; same padding is used for all the conv and max-pooling layers.*

Network
Input: $\mathbf{x} \in \Re^{400\times 60}$
conv (7×7, 16, relu), max-pooling(3×3, stride 2 × 2), BN conv (5×5, 32, relu), max-pooling(3×3, stride 2 × 2), BN conv (3×3, 64, relu), max-pooling(3×3, stride 2 × 2), BN conv (3×3, 64, relu), max-pooling(3×3, stride 2 × 2), BN conv (3×3, 128, relu), max-pooling(3×3, stride 2 × 2), BN conv (3×3, 128, relu), max-pooling(3×3, stride 2 × 2), BN conv (3×3, 256, relu), max-pooling(3×3, stride 2 × 2), BN Flatten()
FC(512, relu), BN FC(512, relu), BN
Output: softmax(10)

Table 3: *Comparison of different systems on four second (or full-length) utterances. (UER %)*

Baseline methods	Valid.	Test
i-vector SVM(RBF) (Avg 7.6s)	-	9.09
i-vector DNN (Avg 7.6s)	-	8.22
RNN(GRU)256x2 (4.0s)	6.63	7.44
biRNN(GRU)256x2 (4.0s)	7.17	7.90
DCNN (4.0s)	2.43	**2.83**

4.3. Results of baseline systems

Table 3 shows the results of i-vector-based approaches with full length utterances and RNN, biRNN, DCNN with four second utterances. The optimal configuration of the i-vector-based DNN was selected by comparing the different numbers of hidden layers and different dropout settings. We compared RNN and biRNN by changing number of GRU layers with dropout setting of 0.0, 0.3 and 0.5. The RNN model obtained best result with two GRUs and dropout 0.3, and biRNN obtained best result with two GRUs and without using dropout setting. Dropout setting was also investigated on the DCNN models, using dropout could not further improve the performance

Table 4: *Results (Duration match or mismatch) on different duration utterances (0.5s, 1.0s, 1.5s and 2.0s) with DCNN models. (UER %)*

Models	Test				
	0.5s	1.0s	1.5s	2.0s	4.0s
Train with 0.5s	**24.01**	-	-	-	-
Train with 1.0s	39.07	**13.18**	-	-	-
Train with 1.5s	65.71	26.27	**8.63**	-	-
Train with 2.0s	73.66	29.94	12.51	**6.87**	-
Train with 4.0s	75.05	45.18	26.54	15.06	**2.83**

Table 5: *Investigation on training student model (two seconds) with teacher model (four seconds).(UER %)*

Methods	λ	L_{kt}	Valid.	Test
Baseline (2.0s)	-	-	6.00	6.87
KD T=3 (2.0s)	0.5	-	5.57	6.02
KD T=5 (2.0s)	0.5	-	4.93	6.05
KD T=7 (2.0s)	0.5	-	5.60	6.66
KD T=3 (2.0s)	0.1	-	5.80	6.26
KD T=3 (2.0s)	0.3	-	4.70	**5.69**
KD T=3 (2.0s)	0.5	-	5.57	6.02
KD T=3 (2.0s)	0.7	-	5.17	5.77
FRKD (2.0s)	0.5	L2 norm	4.70	5.89
FRKD (2.0s)	0.1	L1 norm	4.83	5.67
FRKD (2.0s)	0.3	L1 norm	4.17	**5.28**
FRKD (2.0s)	0.5	L1 norm	4.23	5.33
FRKD (2.0s)	0.7	L1 norm	4.74	5.49

of the DCNN model. Same to the report in previous works [11, 12], the RNN and DCNN models achieved better performance than i-vector-based approaches on our short utterance LID task. Compared with other systems, the DCNN model performed the best on this dataset. We built DCNN models and tested on all target short utterances, i.e., 2.0s, 1.5s, 1.0s and 0.5s. From the results in table 5, we observed that the performance was degraded with the decrease of the duration. By padding on the shorter utterances, we also tested the shorter utterances with longer utterance-based models. The results showed that the performance was further degraded when the duration of training data and test data was mismatched.

4.4. Results of the proposed method

Before examining the proposed method, we did some investigation on two second utterance tasks. Firstly, investigations were done using knowledge distillation loss function, i.e., Eq. 2, based on the proposed teacher-student framework (Fig. 1). The soft labels were obtained using the four second-based DCNN model. We compared different setting of the temperature T and λ in Eq. 2. Compared with the baseline system, the KD method obtained 17.18% relative improvement with UER 5.69%.

In FitNets, L2 norm-based distance metric loss function was used for hidden layer knowledge transfer learning. In this work, we proposed to use L1 norm-based distance metric loss. Compared with L2 norm, L1 norm has parameter selection ability for it tends to produce sparse coefficients in the solution. We compared the performance of L2 norm and L1 norm-based distance metric by fixing λ to 0.5. The experiment results showed that using the L1 norm-based distance metric performed better. In table 5, we also listed the comparison of different setting of λ. The best result was obtained when λ was set to 0.3.

Fig. 2 displays t-SNE [19] scatter plots for feature of the selected hidden layer on the validation data. The teacher model was the four second utterance-based model, and the baseline and student models were built with one second utterances. From this figure, we observed the feature of the teacher model (Fig. 2.b) were more discriminative than the baseline model (Fig. 2.a). The KD and FRKD methods improved the discriminative of the feature. Compared with KD, FRKD obtained more discriminative feature by mimicking the distribution of the teacher model.

We summarized the results of baseline, KD, FRKD and the combination of KD and FRKD on different duration utterance

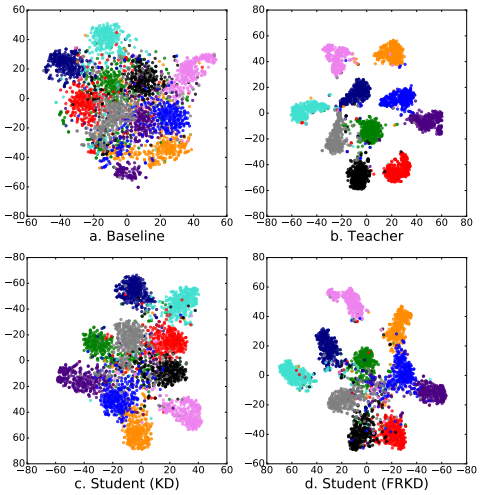

Figure 2: *Feature distribution of the selected hidden layer with t-SNE on validation data.*

Table 6: *Summary of the resutls of baseline, KD, FRKD and combination of KD and FRKD.(UER %)*

Test	Baseline	KD	FRKD	KD+FRKD
Test (2.0s)	6.87	5.69	5.28	4.70
Test (1.5s)	8.63	8.24	7.10	6.42
Test (1.0s)	13.18	13.05	12.12	11.12
Test (0.5s)	24.01	23.41	22.92	21.57

LID tasks in Table 6. For all the student models, we used the same four second-based teacher model, and L1 norm-based distance metric was used. λ was set to 0.3 for both KD and FRKD. T was set to 3. From the results, we observed that both KD and FRKD improved the performance for all the target duration utterances. The proposed method achieved 23.14%, 17.73%, 8.04% and 4.54% relative improvements than the baseline system on 2.0s, 1.5s, 1.0s and 0.5s utterances, respectively, and KD method achieved 17.18%, 4.52%, 0.99% and 2.50% relative improvements than the baseline systems. As the sentences become shorter, the relative improvements become smaller. KD significantly improved the performance on 2.0s utterances, however, for 1.5s, 1.0s and 0.5s utterances, it only had a little improvement. Compared with the KD method, the proposed method performed better on both 2.0s utterances but also 1.5s, 1.0s and 0.5s utterances. Combining KD and FRKD methods further improved the performance. For LID tasks, the experiment results showed that the proposed method is an effective method for improving the performance on short utterances.

5. Conclusions

In this paper, we proposed a teacher-student learning algorithm to explore discriminative feature for short utterances. With the teacher-student network learning, the feature representation knowledge of the long utterances was transferred to the student model to help the student model capturing robust discriminative feature for short utterances. Experiment results showed that the proposed method is an effective method for short duration utterance LID tasks. For future work, we will further investigate on transfer learning for building high performance and small compact model for short utterance LID tasks.

6. References

[1] H. Li, B. Ma and K. A. Lee, "Spoken language recognition: From fundamentals to practice," in Proc. of *The IEEE*, vol. 101, no. 5, pp. 1136-1159, 2013.

[2] C.-H. Lee, "Principles of spoken language recognition," in *Springer Handbook of Speech Processing and Speech Communication*, 2008.

[3] N. Dehak, P. Torres-Carrasquillo, D. Reynolds and R. Dehak, "Language recognition via ivectors and dimensionality reduction," in Proc. of *Interspeech*, pp. 857-860, 2011.

[4] S. O. Sadjadi, J. W. Pelecanos and S. Ganapathy, "Nearest neighbor discriminant analysis for language recognition," in Proc. of *ICASSP*, pp.4205-4209, 2015.

[5] Y. Song, B. Jiang, Y. Bao, S. Wei and L.-R. Dai, "I-vector representation based on bottleneck features for language identification," in *Electronics Letters*, vol. 49, no. 24, pp. 1569-1570, 2013.

[6] P. Shen, X. Lu, L. Liu and H. Kawai, "Local Fisher discrimiant analysis for spoken language identification," in Proc. of *ICASSP*, 2016.

[7] M. Najafian, S. Safavi, P. Weber and M. Russell, "Augmented Data Training of Joint Acoustic/Phonotactic DNN i-vectors for NIST LRE15,", in Proc. of *Odyssey 2016*, June, 2016.

[8] X. Lu, P. Shen, Y. Tsao, H. Kawai, "Pair-wise Distance Metric Learning of Neural Network Model for Spoken Language Identification," in Proc. of *Interspeech*, Sep. 2016.

[9] G. Montavon, "Deep Learning for Spoken Language Identification," *NIPS workshop on Deep Learning for Speech Recognition and Related Applications*, 2009.

[10] S. Ranjan, C. Yu, C. Zhang, F. Kelly and J. Hansen, "Language recognition using deep neural networks with very limited training data," in Proc. of *ICASSP*, 5830-5834, 2016.

[11] I. Lopez-Moreno, J. Gonzalez-Dominguez, O. Plchot, D. Martinez, J. Gonzalez-Rodriguez and P. Moreno, "Automatic language identification using deep neural networks," in Proc. of *ICASSP*, 2014.

[12] A. Lozano-Diez, R. Zazo Candil, J. G. Dominguez, D. T. Toledano and J. G. Rodriguez, "An end-to-end approach to language identification in short utterances using convolutional neural networks," in Proc. of *INTERSPEECH*, 2015.

[13] S. Fernando, V. Sethu, E. Ambikairajah and J. Epps, "Bidirectional Modelling for Short Duration Language Identification," in Proc. of *Interspeech*, 2017.

[14] W. Geng, W. Wang, Y. Zhao, X. Cai and B. Xu, "End-to-End Language Identification Using Attention-Based Recurrent Neural Networks," in Proc. of *Interspeech*, 2016.

[15] G. Hinton, O. Vinyals and J. Dean, "Distilling the knowledge in a neural network," arXiv preprint arXiv:1503.02531, 2015.

[16] A. Romero, N. Ballas, S. E. Kahou, A. Chassang, C. Gatta and Y. Bengio, "Fitnets: Hints for thin deep nets," in Proc. of *ICLR*, 2015.

[17] D. Povey, et al., "The Kaldi speech recognition Toolkit," in proc. of *ASRU*, 2011.

[18] 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," in *Computer Speech & Language*, 2017.

[19] L. Van der Maaten and G. Hinton, "Visualizing data using t-SNE," in *J. Machine Learning Research*, vol.9, pp.2579-2605, 2008.