INTERSPEECH 2016
September 8–12, 2016, San Francisco, USA

Language Identification based on Generative Modeling of Posteriorgram Sequences Extracted from Frame-by-Frame DNNs and LSTM-RNNs

Ryo Masumura, Taichi Asami, Hirokazu Masataki, Yushi Aono, Sumitaka Sakauchi

NTT Media Intelligence Laboratories, NTT Corporation, Japan
{masumura.ryo, asami.taichi, masataki.hirokazu, aono.yushi, sakauchi.sumitaka}
@lab.ntt.co.jp

Abstract

This paper aims to enhance spoken language identification methods based on direct discriminative modeling of language labels using deep neural networks (DNNs) and long short-term memory recurrent neural networks (LSTM-RNNs). In conventional methods, frame-by-frame DNNs or LSTM-RNNs are used for utterance-level classification. Although they have strong frame-level classification performance and real-time efficiency, they are not optimized for variable length utterance-level classification since the classification is conducted by simply averaging frame-level prediction results. In addition, the simple classification methodology cannot fully utilize the combination of DNNs and LSTM-RNNs. To address these issues, our idea is to combine the frame-by-frame DNNs and LSTM-RNNs with a sequential generative model based classifier. In the proposed method, we regard posteriorgram sequences generated from a frame-by-frame classifier as feature sequences, and model them with respect to each language using language modeling technologies. The generative model based classifier does not model an identification boundary, so we can flexibly deal with variable length utterances without loss of conventional advantages. Furthermore, the proposed method can support the combination of DNNs and LSTMs using joint posteriorgram sequences, those of generative modeling can capture differences between two posteriorgram sequences. Experiments conducted using the GlobalPhone database demonstrate the proposed method's effectiveness.

Index Terms: Spoken language identification, DNNs, LSTM-RNNs, generative models, RNNLMs

1. Introduction

Spoken language identification (LID), which automatically determines a language label from input utterances, is a fundamental technique for multilingual speech applications [1, 2]. A lot of technologies have been proposed for LID such as phoneme-based approaches or i-vector based approaches [3, 4]. Additionally, due to recent advances in deep learning (DL) technologies, powerful modeling techniques including deep neural networks (DNNs) or long short-term memory recurrent neural networks (LSTM-RNNs) [5] have been examined for LID fields [6]. Some studies reported that state-of-the-art performance can be obtained by DL technologies.

DL technologies can be split into two main streams for LID application [6]. One of the main streams is the indirect approach; it utilizes DL-based models intended for other usages. Most of these models are trained for predicting sub-phonetic units called senones [7]. Indirect methods are often used for extracting statistics in i-vectors [8] or extracting additional features for other back-end systems [9, 10, 11]. The other main stream is the direct discriminative modeling approach [12, 13, 14, 15]. DL-based models are constructed for directly classifying language labels. This paper focuses on enhancing the direct discriminative modeling approach.

In the direct discriminative modeling approach, frame-by-frame DNNs and LSTM-RNNs are often used [12, 13, 14, 15]. Frame-by-frame models can determine a frame-level posterior probability for each language label. In this case, utterance-level classifiers can be constructed by simply averaging the frame-by-frame prediction results. They offer early determination of the language label at any timing, and so are suitable for real-time applications. In addition, previous studies reported that they deal with short utterances better than i-vector based schemes.

However, there are two issues with the conventional direct discriminative modeling approach. First, frame-by-frame discriminative models are not optimum for variable length utterance-level classification although their frame-level prediction performance is significant. In fact, the utterance-level score was calculated by simply averaging the frame-level prediction scores in previous works. If we directly handle variable-length utterances in discriminative modeling, it is necessary to represent them using compact representations such as i-vectors [16, 17]. Unfortunately, the use of compact representations eliminates the advantage of support for real-time applications since they are often extracted after utterance completion. The second issue is that the conventional methods cannot fully utilize the combination of DNNs and LSTM-RNNs because the simple averaging strategy cannot flexibly handle the differences between predicted sequential results.

Our idea for addressing the issues is to use a sequential generative model based classifier to combine frame-by-frame discriminative models. The generative model based classifier is suitable for variable-length sequences because it does not model identification boundaries. In fact, outputs in the conventional frame-by-frame classifier, i.e., posteriorgrams, can be regarded as feature sequences [18, 19]. By constructing a sequential generative model based classifier using the features, we can expect improved flexibility for variable-length utterances without losing frame-by-frame processing. Furthermore, sequential generative models can use joint sequences to handle multiple sequences in a single framework. Sequential generative modeling is expected to yield a classifier that can capture the differences between individual sequences.

This paper is an initial study that uses a sequential generative model based classifier to integrate frame-by-frame DNNs and LSTM-RNNs. To model the posteriorgram outputs by the model, this paper uses language models (LMs). Thus, we tokenize the posteriorgrams, and model them as an LM with re-

http://dx.doi.org/10.21437/Interspeech.2016-719

spect to language labels. The LM-based classifier will well perform in LID applications since similar ideas are found in traditional phoneme-based approaches [20, 21]. Additionally, we enhance the proposed method with the following two expansions. First, we introduce two state-of-the-art LMs, hierarchical Pitman-Yor LMs (HPYLMs) [22] and RNNLMs [23], for sequential generative modeling. Second, we deal with DNNs and LSTM-RNNs simultaneously by sequential generative modeling of their joint posteriorgrams. In our evaluation, we compare the proposed method with the conventional direct discriminative modeling methods including bidirectional LSTM-RNNs [24] in an identification task.

This paper is organized as follows. Section 2 describes the conventional classification framework based on frame-by-frame DNNs and LSTM-RNNs. In Section 3, we detail novel techniques based on the generative modeling of posteriorgrams extracted from conventional method. Section 4 describes our experiments using the GlobalPhone database [25]. Section 5 concludes this paper.

2. LID using Frame-by-Frame DNNs and LSTM-RNNs

2.1. Direct discriminative modeling

LID is defined as the problem of determining language label $\hat{l}$ from input utterance $\boldsymbol{X} = \boldsymbol{x}_1, \cdots, \boldsymbol{x}_T$, where $\boldsymbol{x}_t$ means an acoustic feature vector of the t-th frame. Actually, since input utterance length T is variable, frame-by-frame discriminative models are often used [12, 13, 14, 15]. In this case, we can compute an utterance-level score by simply averaging the frame-level prediction score. Thus, LID based on frame-by-frame direct discriminative modeling is defined as:

$$\hat{l} = \underset{l \in \mathcal{L}}{\arg\max} \frac{1}{T} \sum_{t=1}^{T} \log P(l|\boldsymbol{X}, t, \boldsymbol{\Theta}), \tag{1}$$

where $\mathcal{L}$ represents a set of target languages, and $\boldsymbol{\Theta}$ is a model parameter of a discriminative model. $P(l|\boldsymbol{X}, t, \boldsymbol{\Theta})$ represents a posterior probability of label l in the t-th frame. This determination can be conducted in an online manner. Thus, it supports determination even before reaching the end of the utterance.

2.2. Frame-by-Frame DNNs and LSTM-RNNs

In order to obtain the frame-level posterior probability, frame-by-frame DNNs and LSTM-RNNs can be used. Each model can be implemented in an online (left-to-right) manner.

DNNs are full-connected feed-forward NNs with multiple hidden layers. When using DNNs, the input layer is composed by stacking a currently-being-processed frame and its left-right contexts. The DNN-based frame-level posterior probability is calculated as:

$$P(l|\boldsymbol{X}, t, \boldsymbol{\Theta}_{\texttt{DNN}}) = P(l|\boldsymbol{i}_t, \boldsymbol{\Theta}_{\texttt{DNN}}), \tag{2}$$

$$\boldsymbol{i}_t = [\boldsymbol{x}_{t-M}^{\top}, \cdots, \boldsymbol{x}_t^{\top}, \cdots, \boldsymbol{x}_{t+M}^{\top}]^{\top}, \tag{3}$$

where M denotes context size in the input layer.

Unidirectional LSTM-RNNs can automatically store previous long-range information in hidden layers without stacking previous frames [5]. The LSTM-based frame-level discriminative probability is calculated as:

$$P(l|\boldsymbol{X}, t, \boldsymbol{\Theta}_{\texttt{LSTM}}) = P(l|\boldsymbol{x}_t, \boldsymbol{h}_{t-1}, \boldsymbol{\Theta}_{\texttt{LSTM}}) \tag{4}$$

where $\boldsymbol{h}_{t-1}$ represents outputs of the previous hidden layers. Note that unidirectional LSTM-RNNs can be fused with DNNs by averaging each log probability [13].

In addition, we can use bidirectional LSTM-RNNs, which can utilize entire utterance information in the hidden layers if batch implementation is allowed. [24]

3. LID based on Sequential Generative Modeling of Posteriorgrams

3.1. Overview

This paper proposes an LID method that combines conventional frame-by-frame DNNs and LSTM-RNNs with sequential generative model based classifier. In the proposed method, we regard outputs of the conventional frame-by-frame classifier, i.e., posteriorgrams, as features. Posterior probabilities in the t-th frame is represented as an $|\mathcal{L}|$-dimensional vector:

$$\boldsymbol{y}_t = [P(l_1|\boldsymbol{X}, t, \boldsymbol{\Theta}), \cdots, P(l_{|\mathcal{L}|}|\boldsymbol{X}, t, \boldsymbol{\Theta})]^{\top}. \tag{5}$$

The proposed method consists of the following steps. First, in a tokenization step, the posteriorgram sequence of target utterance $\boldsymbol{Y} = \boldsymbol{y}_1, \cdots, \boldsymbol{y}_T$ is converted into token sequence $\boldsymbol{S} = s_1, \cdots, s_T$. Next, in a classification step, the tokenized posteriorgram is fed into each LM constructed for each language label, and a language label of the target utterance is determined by calculating the generative probabilities.

3.2. Tokenizer

In order to model the posteriorgram using generative LMs, a tokenizer is necessary. The tokenizer is trained from the posteriorgrams of training data sets. To this end, this paper uses K-means clustering with Euclidean distance. In K-means clustering, K centroids are trained from all of the posteriorgrams in training data sets. The centroids are denoted as $\boldsymbol{c}_1, \cdots, \boldsymbol{c}_K$.

In a tokenization step, a posteriorgram is tokenized in a frame-by-frame manner. Thus, tokenization can be implemented in an online manner. The tokenization of $\boldsymbol{y}_t$ is given by:

$$s_t = \underset{k \in 1, \cdots, K}{\arg\min} D(\boldsymbol{c}_k, \boldsymbol{y}_t), \tag{6}$$

where D denotes the Euclidian distance of two vectors. Thus, the posteriorgram is converted into an index number sequence of the nearest neighbor centroids.

3.3. Classifier

An LID classifier is composed by LMs that are constructed from the tokenized posteriorgrams for each language label. The classification is defined as:

$$\hat{l} = \underset{l \in \mathcal{L}}{\arg\max} P(\boldsymbol{S}|\boldsymbol{\theta}_l), \tag{7}$$

where $\boldsymbol{\theta}_l$ denotes a model parameter of LM for language label l. This determination can be also conducted in an online manner. Thus, the proposed method supports early determination as does the conventional classifier.

The LM for language label l is trained from tokenized posteriorgrams of the target language in the same training data sets as those for the frame-by-frame DNNs and LSTM-RNNs. This paper employs two LMs; HPYLMs and RNNLMs.

HPYLMs are Bayesian n-gram LMs, which are known to be one of the most accurate n-gram LMs [22]. HPYLMs define the generative probability of the t-th token, s_t, given its $N-1$

Table 1: *Experimental data sets: number of utterances.*

	Train		Valid	Test
French (FR)	9,862	(25.3 h)	308	308
German (GE)	9,496	(17.1 h)	284	303
Korean (KO)	7,794	(20.1 h)	153	160
Mandarin (MA)	9,608	(29.3 h)	203	273
Portuguese (PO)	9,568	(24.5 h)	315	256
Russian (RU)	11,549	(24.8 h)	234	269
Shanghai (SH)	2,179	(7.9 h)	137	227
Spanish (SP)	6,500	(20.8 h)	131	202
Swedish (SW)	11,168	(20.4 h)	154	381
Thai (TH)	13,739	(27.3 h)	100	150
Turkish (TU)	6,489	(15.9 h)	121	281
Vietnamese (VI)	18,089	(18.6 h)	231	371
ALL	116,041	(252.0 h)	2,371	3,181

tokens $\boldsymbol{s}_{t-N+1}^{t-1}$. The generative probability of token sequence $\boldsymbol{S}$ is defined as:

$$P(\boldsymbol{S}|\boldsymbol{\theta}_l^{\texttt{HPY}}) = \prod_{t=1}^{T} P(s_t|\boldsymbol{s}_{t-N+1}^{t-1}, \boldsymbol{\theta}_l^{\texttt{HPY}}). \tag{8}$$

RNNLMs are state-of-the-art LMs, and flexibly take long-range context information into consideration based on their recurrent structure [23]. In RNNLMs, the generative probability of token sequence $\boldsymbol{S}$ is defined as:

$$P(\boldsymbol{S}|\boldsymbol{\theta}_l^{\texttt{RNN}}) = \prod_{t=1}^{T} P(s_t|s_{t-1}, \boldsymbol{z}_{t-1}, \boldsymbol{\theta}_l^{\texttt{RNN}}), \tag{9}$$

where $\boldsymbol{z}_{t-1}$ denotes previous the output of the hidden layer. It includes long-range context information while n-gram LM uses only $n-1$ context information.

3.4. Joint posteriorgram

We can deal with multiple posteriorgrams individually generated from different frame-by-frame discriminative models in the same framework by using a joint posteriorgram. A joint posterior-based feature vector is defined as:

$$\bar{\boldsymbol{y}}_t = [\boldsymbol{y}_t^{(1)\top}, \cdots, \boldsymbol{y}_t^{(U)\top}]^\top \tag{10}$$

where U denotes the number of posteriorgrams composing the joint posteriorgram. For example, $\boldsymbol{y}_t^{(1)}$ is extracted from the DNN and $\boldsymbol{y}_t^{(2)}$ is extracted from the unidirectional LSTM-RNN. Joint posteriorgram $\bar{\boldsymbol{Y}} = \bar{\boldsymbol{y}}_1, \cdots, \bar{\boldsymbol{y}}_T$ is tokenized instead of single posteriorgram $\boldsymbol{Y}$.

4. Experiments

4.1. Setups

Our evaluation employed GlobalPhone, a multilingual data corpus [25]. GlobalPhone includes spoken utterances read by native speakers in several languages. The average utterance duration is about 7 seconds. This paper used 12 languages and we split them into training set (Train), validation set (Valid), and test set (Test). Details of the number of utterances and the data size are shown in Table 1.

In our experiments, we used 38 dimensional MFCC coefficients (12MFCC, 12ΔMFCC, 12ΔΔMFCC, Δpower and ΔΔpower) as an acoustic feature; extraction used 20 msec long windows shifted by 10 msec.

Table 2: *Frame-level LID performance: FER (%).*

	Valid	Test
DNN	30.20	35.31
LSTM-RNN	16.41	22.36
BLSTM-RNN	**4.20**	**7.14**

For the evaluations, we constructed three frame-by-frame discriminative models from the training set.

- **DNN**: DNN with 5 hidden layers and 1024 sigmoid units. The input layer was fed with 21 frames formed by stacking the current processed frame and its ±10 left-right context. For training, we used discriminative pre-training to construct an initial network [26], and fine-tuned it using mini-batch stochastic gradient descent (MB-SGD). The validation set was used for early stopping.
- **LSTM-RNN**: Left-to-right unidirectional LSTM-RNN with 3 hidden layers and 512 units. The input was just the target frame without stacking other frames. For training, we used discriminative pre-training to construct an initial network, and fine-tuned it using MB-SGD and back propagation through time algorithm. The validation set was used for early stopping.
- **BLSTM-RNN**: Bidirectional LSTM-RNN with 3 hidden layers and 512 nodes. The training strategy was same as for LSTM-RNN.

These models were used for both the conventional method and the proposed method. Also, we evaluated the combination of DNN and LSTM-RNN.

In addition, we prepared additional components for the proposed method. For constructing a tokenizer, the number of centroids in K-means clustering was set to 64 and 128. The centroids were trained from the posterior sequences of the training data sets. We used two LMs for generative modeling of the tokenized posteriorgrams.

- **HPYLM**: Token-based 3-gram HPYLM. For training, we used 200 iterations for burn-in, and collected 10 samples.
- **RNNLM**: Token-based RNNLM with 200 hidden units. In training, the validation set was used for early stopping.

Both LMs are constructed from tokenized posteriorgrams of the training data set. In these settings, some hyper-parameters such as mini-batch size were adjusted using the validation set.

4.2. Results

First, we investigate the frame-level LID performance of DNN, LSTM-RNN, and BLSTM-RNN since the utterance-level performance is affected by the frame-level performance in both the conventional and proposed methods. Table 2 shows the frame-level error rate (FER) for the validation set and test set.

The results show that DNN was inferior to LSTM-RNN and BLSTM-RNN. The highest performance was attained by BLSTM-RNN since it can use all of the utterance information for determining the target frame. This suggests that frame-level performance depends on whether long-range context information can be used or not.

Table 3: *Utterance-level LID performance: UER (%)*

Conventional methods				Valid				Test			
	Discriminative models			1 sec	2 sec	3 sec	Whole	1 sec	2 sec	3 sec	Whole
(1).	DNN			**7.55**	**2.11**	**0.76**	0.54	11.82	6.32	4.53	3.12
(2).	LSTM-RNN			14.34	4.22	2.45	0.91	16.54	7.11	5.22	2.55
(3).	DNN+LSTM-RNN			9.24	3.04	1.40	**0.38**	**11.67**	**5.19**	**4.00**	2.17
(4).	BLSTM-RNN			-	-	-	0.55	-	-	-	**1.58**
Proposed methods				Valid				Test			
	Discriminative models	LMs	#tokens	1 sec	2 sec	3 sec	Whole	1 sec	2 sec	3 sec	Whole
(5).	DNN	HPYLM	64	7.61	1.95	0.81	0.43	10.45	5.39	3.09	1.51
(6).			128	8.02	2.22	0.85	0.47	11.17	5.51	4.03	2.21
(7).		RNNLM	64	7.68	1.85	0.89	0.48	10.35	5.19	2.33	1.07
(8).			128	8.12	2.12	0.97	0.64	11.35	5.32	3.78	2.02
(9).	LSTM-RNN	HPYLM	64	11.52	3.46	1.52	0.64	13.43	5.25	3.53	1.22
(10).			128	10.94	3.42	1.95	0.68	13.74	5.22	3.68	1.29
(11).		RNNLM	64	10.68	2.63	1.23	0.51	12.05	4.59	3.21	1.14
(12).			128	10.09	2.75	1.40	0.72	11.95	5.16	3.37	1.26
(13).	DNN+LSTM-RNN	HPYLM	64	6.88	1.69	0.68	0.30	9.50	3.59	2.21	0.92
(14).			128	6.75	1.52	**0.64**	0.29	9.34	3.34	1.73	0.63
(15).		RNNLM	64	6.67	1.61	**0.64**	**0.26**	**8.59**	3.05	1.89	0.76
(16).			128	**6.50**	**1.48**	**0.64**	0.30	8.93	**2.99**	**1.51**	**0.50**

Table 4: *Utterance-level LID performance per language label with 3 sec determination using test set: UER (%).*

	FR	GE	KO	MA	PO	RU	SH	SP	SW	TH	TU	VI	ALL
(3).	0.00	**0.00**	0.00	0.74	**0.40**	6.70	23.35	13.87	3.44	0.00	3.56	0.00	4.00
(16).	0.00	0.34	0.00	**0.37**	2.35	**4.84**	**2.21**	**6.44**	**1.32**	0.00	**1.43**	0.00	**1.51**

Next, we investigated utterance-level LID performance using an identification task that evaluates hard decision accuracy by selecting the top scored language; accordingly, utterance-level error rate (UER) was used as the evaluation metric. In addition, for evaluation of early determination performance, we also examined evaluation using the results achieved after 1 sec, 2 sec, and 3 sec. Note that the number of utterances in early determination is the same as in whole-utterance classification.

The results for the conventional methods are shown on lines (1) to (4) of Table 3. LSTM-RNN showed higher performance than DNN in classifying whole utterances. On the other hand, LSTM-RNN was inferior to DNN in the early determination task although LSTM-RNN was quite superior to DNN in terms of frame-level performance. This suggests that frame-level classification performance is not always related to utterance-level classification performance. In addition, the combination of DNN and LSTM-RNN was not always effective for utterance-level classification. In fact, their combination degraded the performance for the validation set. It seems that the conventional combination method that averages both scores is directly affected by individual classification performances. Among the conventional methods, the highest performance was attained by BLSTM-RNN when classifying whole utterances but we note that it cannot be used for real-time applications.

The results of the proposed method are shown on lines (5) to (16) of Table 3. They confirm that the proposed methods that use the generative model based classifier is superior to the conventional methods in most conditions. In addition, RNNLM was superior to HPYLM for sequential generative modeling in most conditions. It seems that the ability of RNNLM to capture long-range context information yielded the improvements. In addition, the combination of DNN and LSTM-RNN showed superior performance to using either DNN or LSTM in isolation in all conditions. This suggests that the proposed method can utilize the combination of different posteriorgrams. The best result was obtained by RNNLM modeling of 128-tokenized joint posteriorgrams extracted from both DNN and LSTM-RNN. The results surpassed those of BLSTM-RNN for whole utterances.

Table 4 shows the best results among the conventional method and those for the proposed method per language label on the test set when classifying is stop after 3 sec. They show that the conventional method was particular inferior to the proposed methods for Spanish and Shanghai. In fact, the conventional method mistook Spanish for Portuguese, and Shanghai for Mandarin. The proposed method could decrease such errors without degrading the performance for other languages.

5. Conclusions

In this paper, we presented a novel method that uses a sequential generative model based classifier to combines frame-by-frame DNNs and LSTM-RNNs. In the proposed method, posteriorgrams generated from conventional frame-by-frame DNNs and LSTM-RNNs are tokenized, and the label determination is conducted by calculating the generative probabilities of tokenized sequences for each per language label. The proposed method matches the real-time efficiency of the conventional method and offers flexibility in terms of utterance length. Also, it can effectively combine DNNs and LSTM-RNNs through the generative modeling of joint posteriorgrams. Experiments using the GlobalPhone database showed that the proposed method offers better LID performance than the conventional method. The best results were obtained by using joint posteriorgrams and an RNNLM based classifier. In future work, we will confirm the proposed method's effectiveness in a verification task.

6. References

[1] T. Niesler and D. Willett, "Language identification and multilingual speech recognition using discriminatively trained acoustic models," *ISCA Workshop on Multilingual Speech and Language Processing*, 2006.

[2] J. Gonzalez-Dominguez, D. Eustis, I. Lopez-Moreno, A. Senior, F. Beaufays, and P. J. Moreno, "A real-time end-to-end multilingual speech recognition architecture," *IEEE Journal of selected topics in signal processing*, vol. 9, pp. 749–759, 2015.

[3] 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.

[4] 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.

[5] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, pp. 1735–1780, 1997.

[6] F. Richardson, D. Reynolds, and N. Dehak, "Deep neural network approaches to speaker and language recognition," *IEEE Signal Processing Letters*, vol. 22, pp. 1671–1675, 2015.

[7] L. Ferrer, Y. Lei, M. McLaren, and N. Scheffer, "Study of senone-based deep neural network approaches for spoken language recognition," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 24, pp. 105–116, 2016.

[8] Y. Lei, L. Ferrer, A. Lawson, M. McLaren, and N. Scheffer, "Application of convolutional neural networks to language identification in noisy conditions," *In Proc. Odyssey*, pp. 287–292, 2014.

[9] L. Ferrer, Y. Lei, M. McLaren, and N. Scheffer, "Spoken language recognition based on senone posteriors," *In Proc. INTERSPEECH*, pp. 2150–2154, 2014.

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

[11] B. Jiang, Y. Song, S. Wei, J.-H. Liu, I. V. McLoughlin, and L.-R. Dai, "Deep bottleneck features for spoken language identification," *PloS one*, vol. 9, no. 7, pp. 1–11, 2014.

[12] I. Lopez-Moreno, J. Gonzalez-Dominguez, and O. Plchot, "Automatic language identification using deep neural networks," *In Proc. ICASSP*, pp. 5337–5341, 2014.

[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 Proc. INTERSPEECH*, pp. 2155–2159, 2014.

[14] J. Gonzalez-Dominguez, I. Lopez-Moreno, P. J. Moreno, and J. Gonzalez-Rodriguez, "Frame-by-frame language identification in short utterances using deep neural networks," *Neural Networks*, vol. 64, pp. 49–58, 2015.

[15] 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, pp. 1–17, 2016.

[16] N. Dehak, P. 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.

[17] N. Dehak, P. A. Torres-Carrasquillo, D. Reynolds, and R. Dehak, "Language recognition via ivectors and dimensionality reduction," *In Proc. INTERSPEECH*, pp. 857–860, 2011.

[18] H. Hermansky, D. P. Ellis, and S. Sharma, "Tandem connectionist feature stream extraction for conventional HMM systems," *In Proc. ICASSP*, pp. 1635–1638, 2000.

[19] G. Chen, C. Parada, and G. Heigold, "Small-footprint keyword spotting using deep neural networks," *In Proc. ICASSP*, pp. 4087–4091, 2014.

[20] M. A. Zissman, "Comparison of four approaches to automatic language identification of telephone speech," *IEEE Transactions on Speech and Audio Processing*, vol. 4, pp. 31–44, 1996.

[21] P. A. Torres-Carrasquillo, D. A. Reynolds, and J. Deller, "Language identification using Gaussian mixture tokenization," *In Proc. ICASSP*, pp. 757–760, 2002.

[22] Y. W. Teh, "A hierarchical Bayesian language model based on Pitman-Yor processes," *In Proc. COLING/ACL*, pp. 985–992, 2006.

[23] T. Mikolov, M. Karafiat, L. Burget, J. Cernocky, and S. Khudanpur, "Recurrent neural network based language model," *In Proc. INTERSPEECH*, pp. 1045–1048, 2010.

[24] A. Graves and J. Schmidhuber, "Framewise phoneme classification with bidirectional LSTM and other neural network architectures," *Neural Networks*, vol. 18, pp. 602–610, 2005.

[25] T. Schultz, N. T. Vu, and T. Schlippe, "GlobalPhone: A multilingual text and speech database in 20 languages," *In Proc. ICASSP*, pp. 8126–8130, 2013.

[26] F. Seide, G. Li, X. Chen, , and D. Yu, "Feature engineering in context-dependent deep neural networks for conversational speech transcription," *In Proc. ASRU*, pp. 24–29, 2011.