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

A Divide-and-Conquer Approach for Language Identification based on Recurrent Neural Networks

G. Gelly[1], *J.L. Gauvain*[1], *V.B. Le*[2], *A. Messaoudi*[2]

[1]LIMSI, CNRS, Univ. Paris-Sud, Université Paris-Saclay, 91405 Orsay, France
[2]Vocapia Research, 91400 Orsay, France

gelly@limsi.fr, gauvain@limsi.fr, levb@vocapia.com, abdel@vocapia.com

Abstract

This paper describes the design of an acoustic language recognition system based on BLSTM that can discriminate closely related languages and dialects of the same language. We introduce a *Divide-and-Conquer* (D&C) method to quickly and successfully train an RNN-based multi-language classifier. Experiments compare this approach to the straightforward training of the same RNN, as well as to two widely used LID techniques: a phonotactic system using DNN acoustic models and an i-vector system. Results are reported on two different data sets: the 14 languages of NIST LRE07 and the 20 closely related languages and dialects of NIST OpenLRE15. In addition to reporting the NIST Cavg metric which served as the primary metric for the LRE07 and OpenLRE15 evaluations, the EER and LER are provided. When used with BLSTM, the D&C training scheme significantly outperformed the classical training method for multi-class RNNs. On the OpenLRE15 data set, this method also outperforms classical LID techniques and combines very well with a phonotactic system.

Index Terms: speech recognition, language identification, RNN, BLSTM

1. Introduction

Automatic spoken language recognition is the task of automatically identifying the language spoken in a given speech segment using the characteristics of the speech signal.

LIMSI has been developing phonotactic systems for language recognition since the early 1990s, when the use of phone-based acoustic likelihoods was proposed for language identification [1, 2]. The basic approach was extended to use parallel phone recognizers with phonotactic characteristics [3], lexical information [4, 5] and phone lattices [6, 7]. Variant approaches based on phone decoding with phonotactic models have been explored for many years and have been shown to provide state-of-the-art results [8, 9, 10].

In preparation for the National Institute of Science and Technology (NIST) 2015 Language Recognition Evaluation (OpenLRE15 [11]), we evaluated acoustic methods that when combined with the phonotactic system improved the language recognition performance [12]. With its innate ability to exploit long range dependencies, Bidirectional Long Short Term Memory (BLSTM) neural networks were natural candidates as purely acoustic classifiers. This choice was also motivated by our previous experience with BLSTM on Speech Activity Detection [13] and some earlier work on this topic [14] which showed good results on short segments for a limited number of languages.

Our first attempt to train multi-class Recurrent Neural Network (RNN) based on LSTM cells gave not-competitive results for language recognition. To overcome this, small RNNs were trained as binary classifiers in order to separate each language from the others. This produced much better results which suggested that the problem was not with the RNN itself or its size but with the training process. To address this problem a four-step training process was designed that we refer to as *Divide-and-Conquer* (D&C). This method was compared with straighforward RNN training. The BLSTM-based system was also compared to two widely employed language identification (LID) techniques: an i-vector system and a phonotactic system.

The next section describes the BLSTM-based language recognizer and the proposed D&C training process. Section 3 provides a short description of the two baseline systems, followed by Section 4 which details the results obtained on two NIST evaluation data sets.

2. RNN-based language classifier

Over the last few years, RNNs and in particular RNNs based on LSTM have been successfully applied to a wide range of classification tasks for which the discriminative information is embedded in a sequence. For spoken language identification, [14] showed that LSTM-RNN can outperform other LID techniques on short utterances for a small number (8) of languages.

This section describes the specific RNN used in this study and a new *divide-and-conquer* training approach to successfully discriminate between the 14 languages of the NIST LRE07 evaluation as well as between the 20 closely related languages of the NIST OpenLRE15 evaluation.

2.1. Augmented BLSTM

Long Short-Term Memory cells as shown in Figure 1 were introduced to overcome some of the shortcomings of classical RNNs [15] and were popularized after Graves demonstrated their good performance for optical character recognition and speech sequence labeling [16, 17].

Given an input sequence $\boldsymbol{p} = (\boldsymbol{p}^1, ..., \boldsymbol{p}^T)$, a standard RNN computes the output vector sequence $\boldsymbol{z} = (\boldsymbol{z}^1, ..., \boldsymbol{z}^T)$ by iterating the equations 1 and 2 from $t = 1 \rightarrow T$:

$$\boldsymbol{h}^t = \sigma_1 \left(\boldsymbol{W}_1 \cdot \tilde{\boldsymbol{p}}^t + \boldsymbol{b}_1 \right) \quad with \quad \tilde{\boldsymbol{p}}^t = \begin{bmatrix} \boldsymbol{p}^t \\ \boldsymbol{h}^{t-1} \end{bmatrix} \qquad (1)$$

$$\boldsymbol{z}^t = \sigma_z \left(\boldsymbol{W}_z \cdot \boldsymbol{h}^t + \boldsymbol{b}_z \right) \qquad (2)$$

The use of LSTM cells instead of the classic summation

http://dx.doi.org/10.21437/Interspeech.2016-180

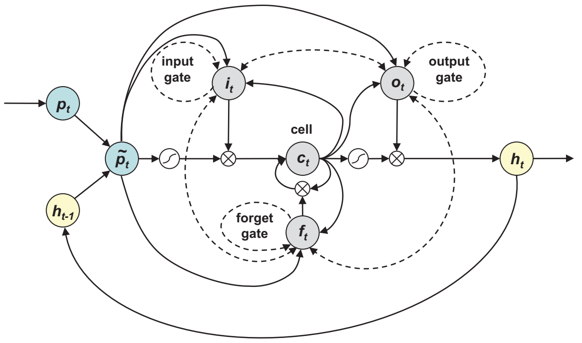

Figure 1: *LSTM cell. The dashed lines correspond to the added links between the gates for the augmented LSTM cell.*

units modifies the computation of $\boldsymbol{h}^t$ as follows:

$$\boldsymbol{i}^t = \sigma_i \left(\boldsymbol{W}_i \cdot \tilde{\boldsymbol{p}}^t + \boldsymbol{W}_i^c \cdot \boldsymbol{c}^{t-1} + \boldsymbol{b}_i \right) \tag{3}$$

$$\boldsymbol{f}^t = \sigma_f \left(\boldsymbol{W}_f \cdot \tilde{\boldsymbol{p}}^t + \boldsymbol{W}_f^c \cdot \boldsymbol{c}^{t-1} + \boldsymbol{b}_f \right) \tag{4}$$

$$\boldsymbol{c}^t = \boldsymbol{f}^t \odot \boldsymbol{c}^{t-1} + \boldsymbol{i}^t \odot \sigma_c \left(\boldsymbol{W}_c \cdot \tilde{\boldsymbol{p}}^t + \boldsymbol{b}_c \right) \tag{5}$$

$$\boldsymbol{o}^t = \sigma_o \left(\boldsymbol{W}_o \cdot \tilde{\boldsymbol{p}}^t + \boldsymbol{W}_o^c \cdot \boldsymbol{c}^t + \boldsymbol{b}_o \right) \tag{6}$$

$$\boldsymbol{h}^t = \boldsymbol{o}^t \odot \sigma_h \left(\boldsymbol{c}^t \right) \tag{7}$$

where $\odot$ is the element-wise multiplication, $\boldsymbol{i}^t$, $\boldsymbol{f}^t$, $\boldsymbol{c}^t$ and $\boldsymbol{o}^t$ are respectively the *input gate*, the *forget gate*, the *cell* and the *output gate* activation vectors. They have all the same size as the hidden vector $\boldsymbol{h}^t$. $\boldsymbol{W}_i^c$, $\boldsymbol{W}_f^c$, and $\boldsymbol{W}_o^c$ are diagonal matrices so that the heart of a cell is only visible to the gates of the same cell.

One shortcoming of conventional RNNs is that they are only able to make use of the left context. For LID purposes there is no reason not to exploit the right context as well. Bidirectional LSTM neural networks (*BLSTM*) were developed to do just that: two distinct LSTM networks process the sequence both forward and backward, and then the output of both networks are combined and fed into the output layers (cf 2.2). This way, one can fully exploit the long range capabilities of LSTM cells. In the literature (e.g. [16, 17]) BLSTM networks are reported to always outperform unidirectional ones, so only BLSTM networks were used in this study.

In [13], an improved version of the LSTM cell was proposed in which direct links are added between the three gates of a cell as shown by the dashed lines in Figure 1.

Equations (3), (4) and (6) are thus modified into (9), (11) and (13):

$$\tilde{\boldsymbol{i}}^t = \boldsymbol{W}_i^i \cdot \boldsymbol{i}^{t-1} + \boldsymbol{W}_i^f \cdot \boldsymbol{f}^{t-1} + \boldsymbol{W}_i^o \cdot \boldsymbol{o}^{t-1} \tag{8}$$

$$\boldsymbol{i}^t = \sigma_i \left(\boldsymbol{W}_i \cdot \tilde{\boldsymbol{p}}^t + \boldsymbol{W}_i^c \cdot \boldsymbol{c}^{t-1} + \tilde{\boldsymbol{i}}^t + \boldsymbol{b}_i \right) \tag{9}$$

$$\tilde{\boldsymbol{f}}^t = \boldsymbol{W}_f^i \cdot \boldsymbol{i}^{t-1} + \boldsymbol{W}_f^f \cdot \boldsymbol{f}^{t-1} + \boldsymbol{W}_f^o \cdot \boldsymbol{o}^{t-1} \tag{10}$$

$$\boldsymbol{f}^t = \sigma_f \left(\boldsymbol{W}_f \cdot \tilde{\boldsymbol{p}}^t + \boldsymbol{W}_f^c \cdot \boldsymbol{c}^{t-1} + \tilde{\boldsymbol{f}}^t + \boldsymbol{b}_f \right) \tag{11}$$

$$\tilde{\boldsymbol{o}}^t = \boldsymbol{W}_o^i \cdot \boldsymbol{i}^t + \boldsymbol{W}_o^f \cdot \boldsymbol{f}^t + \boldsymbol{W}_o^o \cdot \boldsymbol{o}^{t-1} \tag{12}$$

$$\boldsymbol{o}^t = \sigma_o \left(\boldsymbol{W}_o \cdot \tilde{\boldsymbol{p}}^t + \boldsymbol{W}_o^c \cdot \boldsymbol{c}^t + \tilde{\boldsymbol{o}}^t + \boldsymbol{b}_o \right) \tag{13}$$

where the nine matrices $\boldsymbol{W}_{\{i,f,o\}}^{\{i,f,o\}}$ are diagonal so that a gate can only have access to the gates of the same cell.

With these new links the three gates of a cell can interact more efficiently and improve the cell behavior. As a result, this new cell, that we call *LSTM+*, always outperforms classical LSTM cells.

2.2. Network Architecture

The input to the system are 8 PLP coefficients and their first and second derivatives producing 24 dimensional features. They are computed every 10 ms after VTLN is applied. Then, cepstral mean and variance normalization is performed. During our development work, we found that it was beneficial not to process the sequence of features as a whole (whether its duration is 0.5s or 40s) but to truncate it into overlapping sequences of 320 frames ($= 3.2s$) with a shift of 80 frames ($= 0.8s$).

All the RNNs used in this study have the same architecture as shown in Fig. 2. They have two parts: a recurrent network and a feed-forward network that we call the decision network. The recurrent network is composed of two separate LSTM+ networks that process the input sequence in opposite directions. These LSTM+ networks have the same sizes ($c_1 + c_2$ cells) but different weights. The recurrent network produces sequences of vectors with $2 \times c_2$ dimensions that are fed to the decision network. The decision network has only one hidden layer with *tanh* activation functions and a softmax output layer that produces sequences of vectors with o_2 dimensions (one for each language).

Finally, to obtain a single classification vector, the geometric mean of all the output vectors in the output sequences is computed.

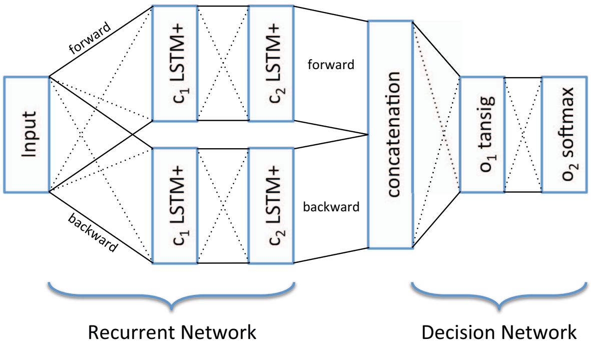

Figure 2: *BLSTM+ Neural Network architecture*

2.3. Divide-and-Conquer (D&C) training

Training of the BLSTM+ neural network was performed using back-propagation through time as described in [18] and a modified version of [16] to take into account the new links we added for BLSTM+. As proposed in [19], the SMORMS3 mini-batch gradient descent algorithm was used as it yielded better results than RMSPROP [20], Adam [21] or Sum of Functions Optimizer [22].

For each training iteration, a small number of training speech segments (about 1000) are randomly selected with an equal number of segments per language to mitigate the effect of the distribution of training segments across languages. During training, we also keep track of the 200 speech segments that lead to the biggest error rates (with an equal number of worst cases per language) and add them to our mini-batch at each training step.

We compare the straightforward training of the multi-class RNN with a four-step process based on D&C strategy:

1. For each language l among the n target languages, we train a small binary classifier to discriminate between l and all the other languages in the training data set. This small binary classifier is a RNN with the architecture described in 2.2 with $c_1 = c_2 = 8$, $o_1 = 2$ and $o_2 = 1$. The total number of weights is then about 8000. Since $o_2 = 1$, we use a *logsig* activation functions for the output layer. Those very small RNNs do not need to be trained extensively: only 200 training iterations per language are performed.
2. The n small RNNs are combined into a larger multi-class classifier. To do so, the weight matrices of the forward and recurrent links of the small RNNs are combined into block diagonal matrices to be used as the weights matrices of the multi-class RNN. Hence, creating n independent channels inside the multi-class RNN, each leading to a single output. The final RNN is similar to the small RNNs but with $c_1 = c_2 = 8 \times n$, $o_1 = 2 \times n$ and $o_2 = n$. For 14 languages, the total number of weights is then about 400000.
3. To balance the impact of the n independent channels inside the recurrent network on the output of our system, we train only the decision network for 100 iterations. During this step, the errors are not back-propagated into the recurrent network and the weights of the *LSTM+* cells are kept constant.
4. Finally, the multi-class RNN is fully trained using the weights obtained at the end of step 3 as a smart initialization point. To improve the behavior of the training, the weights outside the diagonal blocks in the recombined weight matrices are not set to zero exactly but are randomly set using a gaussian distribution with a zero mean and a small variance (10^{-6}).

Classical training of a multi-class RNN consists of performing only the 4th step of the D&C training with a random initialization of the weights.

3. Baseline systems

Two baseline systems were trained: a phonotactic system and an i-vector system. These two classifiers are briefly described here, see [12] for a more detailed description.

3.1. Phonotactic system (PHO)

Phonotactic systems for language identification have been popular since the mid-1990s [1, 2, 3]. Such systems rely on the assumption that the phonotactic characteristics, that is the way phonemes make up words and sentences, differ across languages.

The phonotactic system makes use of the Parallel Phone Recognizer followed by Language Modeling (PPRLM) approach [3]. Pre-trained phone decoders using HMM-DNN acoustic models for three languages (English, Italian and Russian) were used to decode all of the training data. Phone 4-gram statistics are estimated from the resulting phone lattices [6]. The 4-gram statistics are then used to compute the expectation of the phone log-likelihood for each target language. The posteriors of the three phone decoders are averaged, and used as the score for language identification.

3.2. I-vector system (IVC)

The i-vector framework [23] has been successfully applied to both speaker verification [24, 25] and language identification [26].

The i-vector system characterizes languages and utterances with vectors obtained by projecting speech data onto a total variability space T where language and channel information is dense. It is generally expressed as:

$$S = m + Tw \tag{14}$$

where w is called an i-vector and m and S are the GMM super-vector of the language independent UBM and language adapted model, respectively.

During the test phase, the i-vector of the test utterance is scored against the claimant (hypothesized language) specific vector obtained in the training phase, after post-processing the vectors for session variability compensation.

Here, the PLDA (Probabilistic Linear Discriminant Analysis) technique [27], which is also commonly used for speaker verification [24, 25] or gender identification [28], was used.

The i-vector LID system uses 7 MFCC features including C0. Similar to [26], vocal tract length normalization (VTLN) and cepstral mean and variance normalization (CMVN) are applied to both the training and test data. Then the Shifted Delta Coefficients (SDC) [29] are computed and concatenated to the MFCC vector. The final feature vectors have 56 dimensions. The system was implemented using the Kaldi toolkit [30].

4. Results

This section presents results on the closed-set task of NIST LRE07 with its 14 languages and post-evaluation results on the 20 languages of NIST OpenLRE15. Three evaluation metrics are used: the NIST Cavg metric which served as the primary metric for the OpenLRE15 and LRE07 evaluations ([11], [31]), as well as the EER and the average language error rate defined as:

$$\text{LER} = \frac{1}{n_C} \sum_{c \in C} \left(\frac{1}{n_{D_c}} \sum_{d \in D_c} \text{Perr}(d) \right) \tag{15}$$

where C is the set of language clusters, n_C is the number of clusters, D_c is the set of variants for cluster c and n_{D_c} is the number of variants in cluster c (for LRE07, there is only one cluster containing all the 14 target languages).

4.1. LRE07

The goal of the NIST LRE07 closed-set task was to identify the spoken language among 14 target languages: Arabic, Bengali, Chinese (Cantonese, Mandarin, MinNan, and Wu), English (American, Indian), Farsi, German, Hindustani (Hindi, Urdu), Japanese, Korean, Russian, Spanish (Caribbean, Non-caribbean), Thai, Tamil, and Vietnamese. The evaluation data set is composed of 2158 audio files for each of the 3 speech test durations: 3s, 10s and 30s.

Table 1 reports the results obtained on the LRE07 evaluation data set. The two RNNs (with or without D&C training) are identical in size (about 400k weights[1]) and were trained during the same amount of time on the same machines. It can be seen that the D&C training improves the LID results and leads to a

[1] In comparison, the number of parameters used for the i-vector and the phonotactic systems is more than 10^7.

	3 sec			10 sec			30 sec			average		
System	*LER*	*EER*	*CAVG*	*LER*	*EER*	*CAVG*	*LER*	*EER*	*CAVG*	*LER*	*EER*	*CAVG*
PHO	34.53	12.79	18.59	11.66	4.21	6.28	2.48	0.79	1.34	16.22	5.99	8.73
IVC	45.68	18.43	24.60	18.92	8.30	10.19	6.25	3.30	3.36	23.61	10.21	12.72
RNN	46.81	16.41	25.21	20.00	7.29	10.77	9.95	4.22	5.36	25.59	9.73	13.78
RNN-D&C	42.11	15.57	22.67	17.56	6.81	9.45	6.08	3.25	3.28	21.92	9.11	11.80
PHO+IVC	31.76	13.21	17.10	8.84	3.74	4.76	1.74	0.74	0.94	14.12	5.99	7.60
PHO+RNN-D&C	28.40	9.91	15.29	8.44	3.06	4.54	2.41	0.55	1.30	13.08	4.62	7.04

Table 1: Results on LRE07 evaluation data with and without the D&C training are compared to the performance of the baseline systems. The best system combinations are also shown. The combinations results from the geometric mean of the posterior probabilities given by each system.

System	*LER*	*EER*	*CAVG*
PHO	23.5	10.1	15.1
IVC	26.6	10.4	17.4
RNN	30.9	13.4	20.8
RNN-D&C	22.8	8.4	14.6
PHO+IVC	18.6	6.6	11.6
PHO+RNN-D&C	16.2	5.7	10.0

Table 2: Results on the OpenLRE15 evaluation data with and without the D&C training are compared to the performance of the baseline systems. The best system combinations are also shown.

RNN that performs better than the i-vector system. One can see also that both acoustic LID systems perform less well than the phonotactic system. However, combining the RNN-D&C system with the phonotactic system improves the results significantly across all durations and especially on short segments. It also performs better than combining the i-vector system with the phonotactic one.

4.2. OpenLRE15

For the OpenLRE15 evaluation, data were grouped into six language clusters (Arabic, Chinese, English, French, Iberian and Slavic) which contain a total of 20 closely related languages or variants of the same language. As detailed in [12], there was a large mismatch between the official evaluation and training data sets which led to poor results on some of the dialects and made it difficult to analyze and compare the performance of the different systems.

In order to reduce the mismatch, 10% of the files of the evaluation data set were randomly selected and added to the training data. All the LID systems were retrained and tested on the remaining 90% of the evaluation data set. The results of this experiment are given in Table 2.

As observed for the LRE07 data set, the D&C training improves the behavior of the RNN-based LID system (more than 25% gain on the error rates). Moreover, it yields a system that performs better than both the i-vector and the phonotactic systems. As above, combining the two best systems (RNN-D&C and PHO) significantly reduces the error rates.

Figure 3 illustrates the impact of the test segment duration on the performance of each system according to the average LER metric. It can be seen that the performance of the acoustic systems (RNN and IVC) degrades less with decreasing speech duration than the token-based approach (PHO). However, for longer speech durations ($> 20s$), the performance of the phonotactic system is still the best. Figure 3 also shows that the large performance gain brought by combining systems holds for all speech durations.

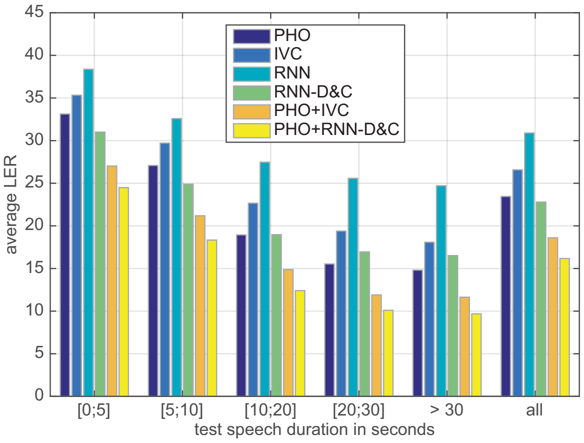

Figure 3: Average LER for the OpenLRE15 test data grouped into intervals according to speech duration of the test speech segments.

5. Conclusions

This paper introduced a *divide-and-conquer* training method that significantly reduced the error rate of a language identification system based on BLSTM-RNNs. This training method has been evaluated on both the NIST LRE07 and NIST OpenLRE15 data sets. In both cases the proposed D&C training significantly outperformed the classical training method for multiclass RNNs.

The resulting RNN LID system was also compared to a phonotactic system and to an i-vector system. The D&C trained RNN outperforms the i-vector system on both data sets and outperforms the phonotatic system on the more challenging OpenLRE15 data set, while requiring an order of magnitude fewer parameters. In addition the D&C trained RNN system combines well with the phonotatic system, leading to the best results by a significant margin compared to any of the three individual systems, for both data sets and for all test segment durations.

6. References

[1] L. Lamel and J.-L. Gauvain, "Identifying non-linguistic speech features." in *Eurospeech*, 1993.

[2] L. F. Lamel and J.-L. Gauvain, "Language identification using phone-based acoustic likelihoods," in *ICASSP*, 1994.

[3] M. A. Zissman *et al.*, "Comparison of four approaches to automatic language identification of telephone speech," *IEEE Transactions on Speech and Audio Processing*, vol. 4, no. 1, p. 31, 1996.

[4] D. Matrouf, M. Adda-Decker, L. Lamel, and J.-L. Gauvain, "Language identification incorporating lexical information." in *ICSLP*, vol. 98, 1998, pp. 181–184.

[5] S. Kadambe and J. Hieronymus, "Language identification with phonological and lexical models," in *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*. IEEE, 1995, pp. 3507–3510.

[6] J.-L. Gauvain, A. Messaoudi, and H. Schwenk, "Language recognition using phone latices." in *INTERSPEECH*, 2004.

[7] D. Zhu and M. Adda-Decker, "Language identification using lattice-based phonotactic and syllabotactic approaches," in *Speaker and Language Recognition Workshop, 2006. IEEE Odyssey 2006: The*. IEEE, 2006, pp. 1–4.

[8] M. F. BenZeghiba, J.-L. Gauvain, and L. Lamel, "Improved n-gram phonotactic models for language recognition." in *INTERSPEECH*, 2010, pp. 2710–2713.

[9] M. F. BenZeghiba, J. Gauvain, and L. Lamel, "Phonotactic language recognition using MLP features," in *INTERSPEECH 2012, 13th Annual Conference of the International Speech Communication Association, Portland, Oregon, USA, September 9-13, 2012*, 2012, pp. 2041–2044. [Online]. Available: http://www.isca-speech.org/archive/interspeech_2012/i12_2041.html

[10] M. F. BenZeghiba, J.-L. Gauvain, and L. Lamel, "Fusing language information from diverse data sources for phonotactic language recognition." in *Odyssey*, 2012, pp. 346–352.

[11] NIST, "The 2015 nist language recognition evaluation plan (lre15)," 2015, http://www.itl.nist.gov/iad/mig//tests/lre/.

[12] G. Gelly, J. Gauvain, L. Lamel, A. Laurent, V. Le, and A. Messaoudi, "Language recognition for dialects and closely related languages," *Odyssey, Bilbao, Spain*, 2016.

[13] G. Gelly and J.-L. Gauvain, "Minimum word error training of rnn-based voice activity detection," *Interspeech*, 2015.

[14] 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." 2014.

[15] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[16] A. Graves, *Supervised sequence labelling with recurrent neural networks*. Springer, 2012, vol. 385.

[17] A. Graves, A.-R. Mohamed, and G. Hinton, "Speech recognition with deep recurrent neural networks," in *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*. IEEE, 2013, pp. 6645–6649.

[18] P. J. Werbos, "Backpropagation through time: what it does and how to do it," *Proceedings of the IEEE*, vol. 78, no. 10, pp. 1550–1560, 1990.

[19] S. Funk, "Rmsprop loses to smorms3," 2015, http://sifter.org/~simon/journal/20150420.html.

[20] T. Tieleman and G. Hinton, "Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude," *COURSERA: Neural Networks for Machine Learning*, vol. 4, p. 2, 2012.

[21] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[22] J. Sohl-Dickstein, B. Poole, and S. Ganguli, "An adaptive low dimensional quasi-newton sum of functions optimizer," *CoRR*, vol. abs/1311.2115, 2013. [Online]. Available: http://arxiv.org/abs/1311.2115

[23] D. Najim, K. Patrick, D. Reda, D. Pierre, and O. Pierre, "Front-End Factor Analysis for Speaker Verification," *Acoustics, Speech and Signal Processing, IEEE Transactions on*, vol. 19, no. 4, pp. 788–798, 2011.

[24] P. Kenny, "Bayesian speaker verification with heavy-tailed priors." in *Odyssey*, 2010, p. 14.

[25] D. Garcia-Romero and C. Y. Espy-Wilson, "Analysis of i-vector length normalization in speaker recognition systems." in *Interspeech*, 2011, pp. 249–252.

[26] D. Martınez, O. Plchot, L. Burget, O. Glembek, and P. Matejka, "Language recognition in ivectors space," *Proceedings of Interspeech, Firenze, Italy*, pp. 861–864, 2011.

[27] S. J. Prince and J. H. Elder, "Probabilistic linear discriminant analysis for inferences about identity," in *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*. IEEE, 2007, pp. 1–8.

[28] S. Ranjan, G. Liu, and J. H. Hansen, "An i-vector plda based gender identification approach for severely distorted and multilingual darpa rats data," in *ASRU*. Scottsdale, AZ: IEEE, 2015.

[29] P. A. Torres-Carrasquillo, E. Singer, M. A. Kohler, R. J. Greene, D. A. Reynolds, and J. R. Deller Jr, "Approaches to language identification using gaussian mixture models and shifted delta cepstral features." in *INTERSPEECH*, 2002.

[30] D. Povey and et al., "The kaldi speech recognition toolkit," in *ASRU*. Hawaii: IEEE, 2011.

[31] NIST, "Nist lre-2007 evaluation plan," 2007, http://www.itl.nist.gov/iad/mig/tests/lre/2007/LRE07EvalPlan-v8b.pdf.