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

Bidirectional Modelling for Short Duration Language Identification

Sarith Fernando[1,2], *Vidhyasaharan Sethu*[1], *Eliathamby Ambikairajah*[1,2], *Julien Epps*[1,2]

[1]School of Electrical Engineering and Telecommunications, UNSW Australia
[2]DATA61, CSIRO, Sydney, Australia

sarith.fernando@unsw.edu.au, v.sethu@unsw.edu.au, e.ambikairajah@unsw.edu.au, j.epps@unsw.edu.au

Abstract

Language identification (LID) systems typically employ i-vectors as fixed length representations of utterances. However, it may not be possible to reliably estimate i-vectors from short utterances, which in turn could lead to reduced language identification accuracy. Recently, Long Short Term Memory networks (LSTMs) have been shown to better model short utterances in the context of language identification. This paper explores the use of bidirectional LSTMs for language identification with the aim of modelling temporal dependencies between past and future frame based features in short utterances. Specifically, an end-to-end system for short duration language identification employing bidirectional LSTM models of utterances is proposed. Evaluations on both NIST 2007 and 2015 LRE show state-of-the-art performance.

Index Terms: Language identification, Short duration utterances, bidirectional LSTM

1. Introduction

Duration mismatch between training and test utterances is a long-standing problem in language identification (LID) [1, 2]. Commonly, long utterances are available for model training but test utterances may be very short during the language recognition phase. This duration mismatch may be compensated by techniques such as shifted delta coefficients (SDC) and eigenfeatures [3]. Most state-of-the-art LID systems rely on the total variability i-vector modelling approach [4] for obtaining fixed length representations of utterances. This elegant framework exhibits low intra-class variability and leads to compact clusters when sufficient statistics can be reliably estimated from an utterance. However, the performance degradation when dealing with short utterances is one of the major challenges in this approach. The i-vector framework has previously addressed short duration utterances by introducing different techniques such as i-vector extraction method using prior distribution [1], and exemplar-based representation [2] for LID. Even though these methods reduce the duration mismatch in i-vector space, the improvements are not significant.

Recently end-to-end automatic LID systems that make use of deep neural networks (DNNs) have been shown to be effective for short duration language identification [5-7]. However, while DNN-based approaches have proven to perform well in several circumstances, they rely on stacking multiple acoustic frames as an input in order to model a longer time context [5]. On the other hand, long short term memory recurrent neural networks have the ability to capture temporal sequences from the connection between units from directed cycles and have become a conventional model when dealing with time dependencies [8]. However, in unidirectional LSTMs, it can be argued that the main disadvantage is that there is no context information concerning future frames.

In the end-to-end modelling approach [7, 9, 10], frame by frame DNN and LSTMs are often used. Frame by frame models can determine a frame-level probability for each language model. In these situations, the language label for a given utterance is computed by identifying the language corresponding to the maximum probability after averaging the frame by frame prediction results. Further, it has been recently shown that averaging only the final 10% of frame-level log probabilities in LSTM networks will lead to better performance compared to averaging all the frame-level log probabilities [8]. However, the underlying assumption used to justify averaging frame-level log probabilities is that the frames are independent of each other, which is not true. The use of recurrent neural network structures, and recently deep bidirectional LSTM (BLSTM) based acoustic models, has been shown to yield the state of the art performance in speech recognition [11-13]. Moreover, [11, 12, 14] show that the performance of this BLSTM performs much better than the unidirectional LSTM and also feed forward neural networks. However, this unified BLSTM mechanism has not been applied to date for short duration in LID.

Motivated by the bidirectional LSTM (BLSTM) mechanism that effectively captures temporal dependencies in the acoustic signal, a bidirectional model is introduced to implement utterance level classification for end to end automatic language identification. Use of BLSTM mechanisms enhance the learning ability in long range discriminative features over the input sequence compered to LSTM for LID systems. Moreover, similar to the i-vector, the proposed utterance level representation gives the ability to successfully extract a fixed length feature vector for an utterance without degrading performance. The underlying idea of this work is to capture the robust discriminative information from short duration utterances for LID. To assess the proposed method, comparisons were conducted between the state-of-the-art i-vector, unidirectional LSTM, and the proposed BLSTM systems and the differences in their performance are shown.

2. Bidirectional LSTM recurrent neural networks

Generally, the memory blocks of the LSTM hidden layers store the temporal state of the input to the network at each time step acting as memory. The probability of a given utterance belonging to one of the identifiable languages is computed at the LSTM output. This result relies on all previous frames from that sequence of the utterance [10, 15].

http://dx.doi.org/10.21437/Interspeech.2017-286

Bidirectional LSTMs are instead based on the idea that the output at time *'t'* may depend on the previous elements in the sequence as well as the future elements. Bidirectional LSTMs are formed by stacking two LSTMs on top of each other as shown in Figure 1. The output is then computed based on the hidden state of both LSTMs. This process is commonly used for tagging and labelling tasks [16, 17], or for embedding a sequence into a fixed length vector. Our purpose is similar to a labeling task in that the language of an utterance must be accurately from a pool of languages [17]. Furthermore, labeling based on the past, present and future samples of the sequence may enhance the predictive capability of the embedded languages, particularly for problems where only small amounts of data are available.

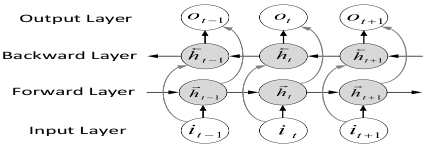

Figure 1: *Structural composition of bidirectional LSTM network using feed-forward and feedback loops.*

The results of the LSTMs in each direction are concatenated in the output layer. In bidirectional LSTM layers, the output y can be computed from the forward sequence $\vec{h}_t$ and backward sequence $\overleftarrow{h}_t$ as

$$\vec{h}_t = \mathcal{H}\left(W_{x\vec{h}}x_t + W_{\vec{h}\vec{h}}\vec{h}_{t-1} + b_{\vec{h}}\right) \quad (1)$$

$$\overleftarrow{h}_t = \mathcal{H}(W_{x\overleftarrow{h}}x_t + W_{\overleftarrow{h}\overleftarrow{h}}\overleftarrow{h}_{t-1} + b_{\overleftarrow{h}}) \quad (2)$$

$$y_t = W_{\vec{h}y}\vec{h}_t + W_{\overleftarrow{h}y}\overleftarrow{h}_t + b_y \quad (3)$$

where x_t, $\vec{h}_t$ and $\overleftarrow{h}_t$ are the acoustic feature input and the two hidden states respectively at time t. For each LSTM memory block, the recurrent hidden layer function $\mathcal{H}$ is derived in the conventional manner [11]. The language identification process trains parameters of the proposed neural network system (Section 3.4) using a supervised approach for the target language, so that the system provides language aware alignments. Based on its structure, this bidirectional model may benefit in capturing high level phoneme information from cell weights. Forget weights may help to reduce other common variations between languages when trained on short duration data. Moreover, unlike i-vectors, capturing sequential information could help to mitigate the duration mismatch for short duration utterances.

3. Experimental Setup

3.1. Databases and Evaluation measures

In order to perform this comparison, the data provided by the National Institute of Standards and Technologies (NIST) in 2007 and 2015 for Language Recognition Evaluations (LRE) was used. The NIST LRE 2007 dataset [18] was used for demonstrating the effectiveness of the proposed BLSTM based model adopted in this paper. The test corpus is a 3s condition evaluation set from NIST LRE 2007. There are 14 languages and 2158 segments included in the 3s evaluation data. The amounts of training data ranged from 264 hours for English to a mere 1.45 hours for Thai.

Another dataset used for the experiments is NIST LRE 2015 [19]. This dataset contains limited training data from conversational telephone speech (CTS) and broadcast narrowband speech (BNBS). The dataset includes 20 languages grouped according to 6 different clusters. The total amount of data in each language varies from 30 minutes to over 100 hours. The evaluation set contains 33784 segments of 3s condition data in the 20 languages included in NIST LRE 2015.

In all experiments, training data set was split into 3s chunks and 15% of the training data was held out as a development set. Two different metrics were used for performance evaluation. The accuracy, i.e. the percentage of correctly identified trials when making hard decisions based on the maximum probability for each target language, was calculated and the Equal Error Rate (EER) was computed language by language for both NIST 2007 and 2015 databases.

3.2. Bottleneck Feature Extraction

The sequential input of the BLSTM-based model was a 42-dimensional vector of acoustic bottleneck features (BNFs). BNFs were extracted using a DNN trained on MFCC features fixing the output layer to denote triphones (4199-state senone). MFCC features were extracted with a 10ms frame shift from 25ms windows. The DNN was trained on 300 hours of Switchboard 1 data. The DNN consisted of 5 layers each with 1024 nodes except at the bottleneck layer (4th layer). All of these layers used a $tanh$ activation function with the exception of the bottleneck layer. The bottleneck layer comprised 42 nodes using a linear activation function. After extracting bottleneck features, vector quantization voice activity detection (VQ-VAD) was used. Compared with MFCC features, these BNFs contain phonetically rich information and have been shown to provide outstanding performance gains compared with typical i-vector systems [20].

3.3. Reference i-vector system

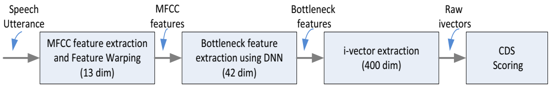
Speech Utterance
MFCC features
Bottleneck features
Raw ivectors
MFCC feature extraction and Feature Warping (13 dim)
Bottleneck feature extraction using DNN (42 dim)
i-vector extraction (400 dim)
CDS Scoring

Figure 2: *Block diagram of reference i-vector system.*

The reference i-vector system shown in Figure 2 follows the standard procedure of [20] and is built on 42-dimensional BNFs. The universal background model (UBM) consisting of 1024 Gaussian components was trained and the total variability subspace of 400 dimensions was derived for this UBM with 10 EM iterations. Simple cosine distance scoring (CDS) was performed to classify these i-vectors, after projecting them to a lower dimension space based on linear discriminant analysis (LDA).

3.4. Proposed BLSTM system description

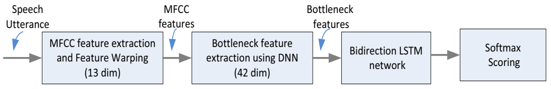
Speech Utterance
MFCC features
Bottleneck features
MFCC feature extraction and Feature Warping (13 dim)
Bottleneck feature extraction using DNN (42 dim)
Bidirection LSTM network
Softmax Scoring

Figure 3: *Block diagram of the proposed end to end bidirectional LSTM system.*

For experimental comparison, LSTM and bidirectional LSTM based frame level LID systems were established as in Figure 3. All models were trained with the truncated back-propagation through time (BPTT) algorithm [21, 22]. The overall system contained two bidirectional LSTM layers followed by two fully connected layers. The third hidden layer used a linear activation function while the fourth hidden layer used rectified linear units (ReLUs). 42 dimensional BNFs were used as the network's input, as explained in Section 3.2. Each hidden layer contained 512 nodes except the third layer, which had only 400 nodes. This 400 nodes hidden layer was used for feature extraction for analysis purposes. The output was a softmax layer with the same number of units as a number of languages and used the ADAM optimizer as in [23] along with cross entropy error for the back propagation. A LSTM system was established for comparison purposes using same above system configuration except the two bidirection LSTM layers were replaced by LSTM layers.

4. Results and Analysis

4.1. Utterance-level representation

In this work, two types of utterance level representation methods were investigated for both LSTM and bidirectional LSTM networks on the NIST LRE 2007 3 second condition. As shown in Figure 4, the method of training on frame-level labels and predicting test labels by averaging the final 10% of log probabilities (*Average),* was compared with the proposed method of training on frame level labels and predicting test labels on only the final frame probability (*Final_frame*). This proposed method may be a more successful approach compared to averaging the frame level predictions, or an i-vector approach, which does not account for any sequential information.

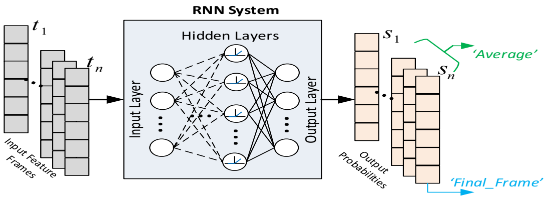

Figure 4: *Comparison of utterance level representation methods for RNN systems.*

Table 1: *Comparison of the performance of LSTM and BLSTM for utterance level representation tested on NIST LRE 2007 (3 sec. cond.)*

Method	Accuracy (EER)	
	LSTM	BLSTM
Average	58.39 (29.74)	**64.37 (25.39)**
Final_frame	58.20 (18.77)	**64.23 (15.64)**

The underlying ideas behind both methods are similar: the frames belonging to a particular language are close to each other in the language embedding space and this will provide an utterance representation that makes use of frame clustering. According to the initial evaluations shown in Table 1, neither LSTMs nor BLSTMs showed significant performance improvement by using the '*Average*' scoring method compared to the '*Final_frame'* method. The main reason could be that there are more fully connected layers followed by RNN layers as described in Section 3.4, and these fully connected layers can broadly capture the variation among sequential information. However, it is noticeable that the EER is degraded in the '*Average*' method compared with the '*Final_frame'* method. Therefore, frame-to-frame training and prediction of test labels on the only final frame (*Final_frame*) in an utterance were used for further experiments.

4.2. Analysis of feature space

Prior to evaluation of the performance of the proposed LID system, the feature space of each system was investigated to study feature reliability. In order to visualize feature distributions, the t-SNE [24] algorithm was used to project the feature vectors into two dimensional space (t-SNE-map) using 400-dimensional test and training vectors together for each system (see Figure 5). t-SNE retains local similarities between samples in the two-dimensional space at the cost of retaining the similarities between dissimilar samples non linearly unlike to PCA and MDS that use the same linear mapping to all data.

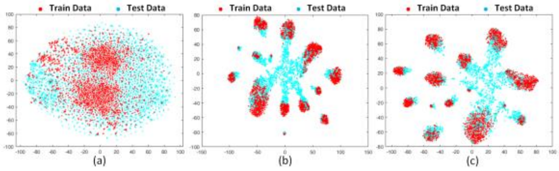

Figure 5: *Illustration of t-SNE feature maps for (a) i-vector, (b) LSTM and (c) BLSTM feature vector*

Figure 5 suggests why both LSTM and bidirectional LSTM systems may perform better compared with i-vectors. Even though t-SNE is a non-parametric algorithm it is able to preserve language cluster information from 400 dimension feature vectors in both LSTM and bidirectional LSTM systems (14 separate clusters). However, this clustering ability is not clearly seen in for the i-vector case, even after applying LDA. When comparing LSTM and bidirectional LSTM it can be noticed that the LSTM feature map is more scattered compared to bidirectional LSTM and the bidirectional LSTM t-SNE-map shows a better training (red) and test (blue) distribution match. In order to demonstrate this argument, the *J*-measure [3] was also computed in both the original feature spaces of training and test sets. The J-measure is the ratio between inter-class scatter to intra-class scatter and larger the value of *J*-measure, the better the discrimination of the classes in the feature space.

Table 2: *Comparison of J-measure on training and test sets for the proposed BLSTM system with the LSTM and reference i-vector system tested on NIST LRE 2007 (3 sec. cond.)*

J- measure	On training set	On test set
i-vector	10.82	5.15
LSTM	**12.29**	6.34
BLSTM	11.87	**6.70**

Table 2 gives the *J*-measure evaluated on both the training and test sets, and it can be seen that LSTMs and BLSTMs lead to better discrimination between languages compared to i-vectors when modelling the same feature space. LSTM features also have a higher *J*-measure on the training set; on the test set this is dominated by BLSTM features. This may be a pre-indications that BLSTMs perform better than LSTMs. Both t-SNE maps and *J*-measure show the effectiveness and sensitivity of the BLSTM feature compared with the reference i-vector and LSTM system.

4.3. Reliability and effectiveness of BLSTM mechanism

The main difference between LSTMs and BLSTMs is in the structural composition. Specifically, the BLSTM contains an additional feedback node. The corresponding frame-level classification score on the target category was analyzed to compare the reliability and effectiveness of the bidirectional mechanism compared with LSTM. For a number of utterances, the frame-level probability score to the target language was analyzed. This showed that the sequential output of the RNNs (both LSTMs and bidirectional LSTMs) becomes more discriminative over time based on the modelling ability of long span dependencies. However, the LSTM classification score took time to reach a high probability value, whereas the BLSTMs achieved high probability scores from the start. The higher initial score that is maintained uniformly confirms the better effectiveness and reliability of the BLSTM mechanism over the LSTMs.

Performing eigenvalue decomposition enables us to explicitly find the orthogonal directions of maximal variations in the data through eigenvectors and eigenvalues. Eigenvalues are directly related to the proportion of information captured along the directions of the corresponding eigenvectors. The capacity of BLSTMs and LSTMs to learn language information embedded in short duration speech signals was explored by observing the eigenvalue variation of the cell and forget weights from layer 1 in each system. Figure 6 shows that BLSTM (red) captures more information (has higher eigenvalues) compared to LSTMs (blue), when systems are trained for short duration utterances. Thus, the cumulative sum of the eigenvalues are higher in BLSTMs than in LSTMs.

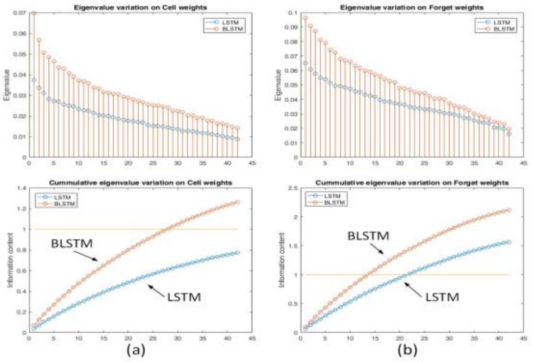

Figure 6: *Language information embedded in (a) cell weights and (b) forget weights from eigenvalue and cumulative eigenvalue variations.*

Table 3 gives a comparison of the LSTM and the BLSTM end-to-end LID systems for both direct decision score (softmax probability) and cosine distance score (CDS) calculation approaches. The direct scoring in LSTM and BLSTM-based LID systems is identical to the method '*Final_Frame'* adopted from Section 4.1. The CDS scoring in LSTM and BLSTM LID systems were calculated for the 400-dimension feature vectors extracted from the linear layer of each system. These vectors are identical to i-vectors and only final feature vector for each utterance was extracted. First, it was shown that the i-vector system performs better for short duration only if the system was trained from long duration utterances. This is expected since the i-vector extraction depends on statistics. Consequently, the reliability of these statistics increase with the utterance duration.

The BLSTM-based end-to-end (direct scoring) LID system seen in Table 3 achieves a relative improvement of 18.69% in terms of accuracy and 16.99% relative EER reduction compared with the referenced i-vector system. This excellent performance confirms the effectiveness of the BLSTM mechanism. Moreover, it is clear that end-to-end LSTM and bidirectional LSTM systems perform similarly to CDS for the extracted features from all systems. Finally, linear logistic regression fusion using the FoCal Multiclass toolkit [25] was applied to the two individual systems described for both LSTM and BLSTM systems with the i-vector system. The result of this fusion is better than the systems itself.

However, it is noticeable that the fused BLSTM extracts less complementary information compared to the fusion i-vector system trained on long duration utterances.

Table 3: *Performance of the proposed system compared to reference i-vector and LSTM systems for NIST LRE 2007 3s condition.*

LID System	Accuracy (EER)	
	Direct	CDS
i-vector (trained on short duration)	-	32.81 (27.77)
i-vector (trained on long duration) (i-vec long)	-	52.22 (18.84)
LSTM	58.20 (18.77)	58.48 (17.56)
BLSTM	64.23 (15.64)	**64.83 (18.38)**
i-vec long + LSTM	64.32 (15.80)	65.48 (14.27)
i-vec long + BLSTM	66.87 (15.24)	**68.54 (12.92)**

The results for NIST 2015 LRE are shown in Table 4. Interestingly and unlike the NIST 2007 results, it can be seen that both the LSTM and BLSTM systems failed to outperform the i-vector system. The main difference between the two LRE data sets is that there is a large channel mismatch between training and test data distributions in LRE 2015, which caused both systems to degrade in performance. However, the fusion of i-vector system with LSTM and BLSTM systems independently did outperform the reference i-vector system.

Table 4: *Performance of the proposed system compared to reference i-vector and LSTM systems for NIST LRE 2015 3s condition.*

LID System	Accuracy (EER)
i-vector	33.83 (28.00)
LSTM	24.11 (37.17)
BLSTM	29.40 (34.79)
i-vector + LSTM	37.25 (25.29)
i-vector + BLSTM	**38.39 (25.05)**

5. Conclusion

In this work, a detailed analysis of the use of bidirectional LSTMs for short utterance automatic language identification (LID) has been presented. Specifically, it was shown that a fixed-length feature vector similar to an i-vector can be successfully obtained from BLSTMs for an utterance. The t-SNE maps and *J*-measure suggest a higher clustering ability in the BLSTM feature space relative to i-vectors. Results show that the proposed system clearly outperforms the reference i-vector based system on the NIST 2007 dataset, which is more challenging in terms of its highly unbalanced datasets and its inclusion of close related languages. The BLSTM system gives comparable results for NIST 2015 LRE, and future work will require more attention to deal with the channel mismatch found in this dataset. Finally, it was shown that BLSTMs have a higher capability to capture discriminative information compared with LSTMs and i-vectors for short duration utterances.

6. References

[1] R. Travadi, M. Van Segbroeck, and S. S. Narayanan, "Modified-prior i-vector estimation for language identification of short duration utterances," in *INTERSPEECH*, 2014, pp. 3037-3041.

[2] M.-G. Wang, Y. Song, B. Jiang, L.-R. Dai, and I. McLoughlin, "Exemplar based language recognition method for short-duration speech segments," in *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*, 2013, pp. 7354-7358.

[3] S. Fernando, V. Sethu, and E. Ambikairajah, "Eigenfeatures: An alternative to Shifted Delta Coefficients for Language Identification," presented at the SST2016, Parramatta, Australia, 2016.

[4] 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.

[5] I. Lopez-Moreno, J. Gonzalez-Dominguez, O. Plchot, D. Martinez, J. Gonzalez-Rodriguez, and P. Moreno, "Automatic language identification using deep neural networks," in *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*, 2014, pp. 5337-5341.

[6] A. Lozano-Diez, J. Gonzalez-Dominguez, R. Zazo, D. Ramos, and J. Gonzalez-Rodriguez, "On the use of convolutional neural networks in pairwise language recognition," in *Advances in Speech and Language Technologies for Iberian Languages*, ed: Springer, 2014, pp. 79-88.

[7] A. Lozano-Diez, R. Zazo Candil, J. González Domínguez, D. T. Toledano, and J. Gonzalez-Rodriguez, "An end-to-end approach to language identification in short utterances using convolutional neural networks," in *Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH*, 2015.

[8] R. Zazo, A. Lozano-Diez, and J. Gonzalez-Rodriguez, "Evaluation of an LSTM-RNN System in Different NIST Language Recognition Frameworks," *Odyssey 2016,* pp. 231-236, 2016.

[9] 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.

[10] 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.

[11] A. Zeyer, R. Schlüter, and H. Ney, "Towards online-recognition with deep bidirectional LSTM acoustic models," *Interspeech, San Francisco, CA, USA,* 2016.

[12] H. Sak, A. Senior, and F. Beaufays, "Long short-term memory based recurrent neural network architectures for large vocabulary speech recognition," *arXiv preprint arXiv:1402.1128,* 2014.

[13] A. Zeyer, P. Doetsch, P. Voigtlaender, R. Schlüter, and H. Ney, "A comprehensive study of deep bidirectional LSTM RNNs for acoustic modeling in speech recognition," *arXiv preprint arXiv:1606.06871,* 2016.

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

[15] W. Geng, W. Wang, Y. Zhao, and X. C. a. B. Xu, "End-to-End Language Identification Using Attention-Based Recurrent Neural Networks," presented at the Interspeech, San Francisco, CA, USA, 2016.

[16] P. Wang, Y. Qian, F. K. Soong, L. He, and H. Zhao, "A unified tagging solution: Bidirectional LSTM recurrent neural network with word embedding," *arXiv preprint arXiv:1511.00215,* 2015.

[17] X. Ma and E. Hovy, "End-to-end sequence labeling via bi-directional lstm-cnns-crf," *arXiv preprint arXiv:1603.01354,* 2016.

[18] A. F. Martin and A. N. Le, "NIST 2007 Language Recognition Evaluation," in *Odyssey - The Speaker and Language Recognition Workshop*, 2008.

[19] "The 2015 NIST Language Recognition Evaluation Plan (LRE15)," 2015.

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

[21] A. Graves, "Generating sequences with recurrent neural networks," *arXiv preprint arXiv:1308.0850,* 2013.

[22] 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 *Interspeech*, 2014, pp. 2155-2159.

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

[24] L. v. d. Maaten and G. Hinton, "Visualizing data using t-SNE," *Journal of Machine Learning Research,* vol. 9, pp. 2579-2605, 2008.

[25] "FoCal, Toolkit for Evaluation, Fusion and Calibration of statistical pattern recognizers http://sites.google.com/site/nikobrummer/focal," 2008.