IMPROVED PHONOTACTIC LANGUAGE RECOGNITION BASED ON RNN FEATURE RECONSTRUCTION

Wei-Wei Liu[1], *Wei-Qiang Zhang*[1], *Yongzhe Shi*[1], *An Ji*[2], *Jiaming Xu*[3], *Jia Liu*[1]

[1]Tsinghua National Laboratory for Information Science and Technology
[1]Department of Electronic Engineering, Tsinghua University, Beijing 100084, China
[2]Department of Electrical and Computer Engineering, Marquette University, U.S.A
[3]University of Chinese Academy of Sciences, Beijing 100190, China
liu-ww10@hotmail.com

ABSTRACT

Nowadays phone recognition followed by support vector machine (PR-SVM) has been proposed in language recognition tasks and shown encouraging results. However, it still suffers from the problems such as the curse of dimensionality led by the increasing order of the N-gram feature supervector, the fast increasing number of possible parameters because of fast exact match of the phoneme history, etc.. These problems hamper the capability of N-gram vector space model (VSM) of handling long-term contexts. In this paper, a recurrent neural networks (RNN) based feature reconstruction (FR) method is presented to compensate for the deficiency of the N-grams feature for phonotactic language recognition in this paper. Experiments are implemented on 2009 National Institute of Standards and Technology language recognition evaluation (NIST LRE) database. The results show that the proposed method gives 8.76%, 3.82%, 11.93% relative error rate reduction for 30s, 10s, 3s respectively comparing with the baseline system.

Index Terms— language recognition, recurrent neural networks (RNN), feature reconstruction (FR)

1. INTRODUCTION

Language recognition plays an important role in many applications, such as machine translation, multilingual speech recognition by identifying a language from an spoken utterance [1]. Nowadays, phonotactic language recognition systems [2] and acoustic language recognition systems [3] are two broad kinds of language recognition systems which have been widely used with encouraging results [4]. However, current techniques of phonotactic language recognition system still have limitations. The basic reason is that N-grams feature supervector can not describe relationships for long phonemes sequence effectively [5]. First, the high order N-grams are able to describe long term context more accurately than the lower ones but computationally expensive. Second, the training data used to estimate parameters of high-order N-gram vector space model (VSM) will be never enough. So tri-grams or four-grams are still commonly used to build VSM in practical. Third, many histories of the N-grams are similar, but N-gram VSM assumes exact match of the histories. N-grams feature requires much more parameters to be estimated than actually needed, which sometimes makes VSM not robust enough.

This project is supported by National Natural Science Foundation of China (No. 61005019, No.61273268 and No. 61370034).

While the characteristics of neural networks (NN) indicate that they can be a compensation for the deficiency of the N-gram VSM. The recurrent neural network (RNN) is a refined form of NN for the task of language modeling, which is able to handle long-term contexts since the input vector contains not only the current word but also the previous output from the neurons in the hidden layer [6]. RNN outperforms traditional language models such as N-grams which only contain very limited histories. One reason is its ability to handle longer contexts by clustering similar sparse histories into continuous low-dimensional spaces. Similar histories sharing the same parameter requires less parameters to be estimated from the training data, so the model is more robust. The distributions of N-grams language model and RNN language model have been shown to be complementary [7], we can integrate them together in language recognition system to get better performance. In this paper, we propose an RNN based feature reconstruction system for language recognition.

The rest of the paper is organized as following. In section 2, we review the traditional phonotactic language recognition system. In section 3, the feature reconstruction language recognition system is fully explained. Experimental setup is described in section 4. In section 5 we give our experiments for evaluating the proposed approach. Finally section 6 concludes this paper.

2. BASELINE SYSTEM

In this work we use phone recognition followed by support vector machine (PR-SVM) [8] language recognition system as baseline system. Generally, the language recognition system maps the input data x to a high dimensional feature supervector as following:

$$\mathbf{\Phi} : x \rightarrow \varphi(x). \tag{1}$$

Then the supervector $\varphi(x)$ is sent to the classifier and a decision is made based on the output of the classifier [9]. In this paper

$$\varphi(x) = [p(d_1|\ell_x), p(d_2|\ell_x), ..., p(d_F|\ell_x)], \tag{2}$$

here $F = f^N$ (f is the number of the phonemes of the front-tend phone recognizer and N is the order of N-gram) and $d_i = s_i...s_{i+n-1}$ $(n = N)$ is the N-gram phoneme string. ℓ_x denotes the lattice generated from data x by a phone recognizer. $p(d_i|\ell_x)$ is the probability of the N-gram d_i in the lattice.

In PR-SVM language recognition system an SVM is employed as the classifier, the output score is computed as following:

$$f(\varphi(x)) = \sum\nolimits_l \alpha_l K_{\mathrm{TFLLR}}(\varphi(x), \varphi(x_l)) + d, \tag{3}$$

here $\varphi(x_l)$ are support vectors. K_{TFLLR} is a term frequency log-likelihood ratio (TFLLR) kernel computed as [10]:

$$K_{\mathrm{TFLLR}}(\varphi(x_i),\varphi(x_j)) = \sum_{q=1}^{F} \frac{p(d_q|\ell_{x_i})}{\sqrt{p(d_q|\ell_{\mathrm{all}})}} * \frac{p(d_q|\ell_{x_j})}{\sqrt{p(d_q|\ell_{\mathrm{all}})}}, \quad (4)$$

the $p(d_i|\ell_{\mathrm{all}})$ is the observed probability of d_i across all lattices. In this work the training stage is always carried out with a one-versus-rest strategy.

3. FEATURE RECONSTRUCTION LANGUAGE RECOGNITION SYSTEM

3.1. RNN feature reconstruction

Recurrent neural network language models (RNNLMs) have been recently shown to improve perplexity and error rates compared to traditional n-gram approaches in speech recognition systems [11]. So here we introduce the RNN into language recognition to reconstruct N-gram feature. Figure 1 illustrates the process of RNN feature reconstruction. Here $\mathbf{V}$ and $\mathbf{U}$ are the weights matrix between input and hidden layer and between hidden and output layer respectively:

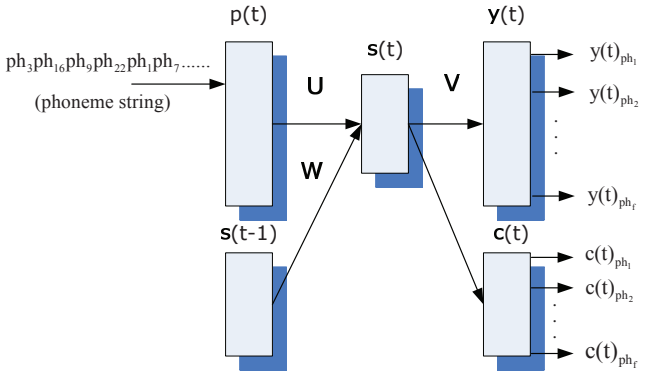

Fig. 1. Process of RNN feature extraction.

$$\mathbf{U}(t_u) = \begin{bmatrix} g_1(\mathrm{ph}_1) & g_2(\mathrm{ph}_1) & \cdots & g_h(\mathrm{ph}_1) \\ g_1(\mathrm{ph}_2) & g_2(\mathrm{ph}_2) & \cdots & g_h(\mathrm{ph}_2) \\ & \vdots & & \\ g_1(\mathrm{ph}_f) & g_2(\mathrm{ph}_f) & & g_h(\mathrm{ph}_f) \end{bmatrix}, \quad (5)$$

$$\mathbf{V}(t_u) = \begin{bmatrix} h_1(\mathrm{ph}_1) & h_1(\mathrm{ph}_2) & \cdots & h_h(\mathrm{ph}_f) \\ h_2(\mathrm{ph}_1) & h_2(\mathrm{ph}_2) & \cdots & h_h(\mathrm{ph}_f) \\ & \vdots & & \\ h_h(\mathrm{ph}_1) & h_h(\mathrm{ph}_2) & & h_h(\mathrm{ph}_f) \end{bmatrix}, \quad (6)$$

where ph_i the ith phoneme of the phone recognizer, t_u is the ultimate time of the training for a single utterance. And the RNN feature is built as:

$$\mathbf{\Phi}_{\mathrm{RNN}} : x \rightarrow \varphi_{\mathrm{RNN}}(x) = [\varphi(x), \varphi_{\mathrm{r}}(\mathrm{ph}_1), \varphi_{\mathrm{r}}(\mathrm{ph}_2), ..., \varphi_{\mathrm{r}}(\mathrm{ph}_f)], \quad (7)$$

where $\varphi_{\mathrm{r}}(\mathrm{ph}_i)$ is a 2*z_h dimensional vector, which is the RNN projection for ith phoneme as following:

$$\varphi_{\mathrm{r}}(\mathrm{ph}_i) = [\mathbf{U}(t_u)_{i1}, \mathbf{U}(t_u)_{i2}, ..., \mathbf{U}(t_u)_{iz_h}, \mathbf{V}(t_u)_{1i}, \mathbf{V}(t_u)_{2i}, ..., \mathbf{V}(t_u)_{z_h i}], \quad (8)$$

here z_h is the size of hidden layer, $\mathbf{V}$ and $\mathbf{U}$ are learned during the training phase as

$$\mathbf{V}(t+1) = \mathbf{V}(t) + \mathbf{s}(t)\mathbf{e}_0(t)^T\alpha, \quad (9)$$

$$\mathbf{U}(t+1) = \mathbf{U}(t) + \mathbf{p}(t)\mathbf{e}_h(t)^T\alpha, \quad (10)$$

where $\mathbf{p}(t)$ denotes the input phoneme of the RNN system. Its dimension equals the size of the phone inventory for a single phone recognizer. $\mathbf{s}(t)$ is the output value from neurons in the hidden layer which contains the state of the network and computed as following:

$$\mathbf{s}(t) = f(\mathbf{U}\mathbf{p}(t) + \mathbf{W}\mathbf{s}(t-1)), \quad (11)$$

where $f(z)$ is sigmoid activation functions:

$$f(z) = \frac{1}{1+e^{-z}}, \quad (12)$$

and the recurrent weights $\mathbf{W}$ are updated as

$$\mathbf{W}(t+1) = \mathbf{W}(t) + \Sigma_{z=0}^{T}\mathbf{s}(t-z-1)\mathbf{e}_h(t-z)^T\alpha, \quad (13)$$

and $\mathbf{e}_0(t)$ is computed using a cross entropy criterion, which denotes the gradient of the error vector in the output layer:

$$\mathbf{e}_0(t) = \mathbf{d}(t) - \mathbf{y}(t), \quad (14)$$

where the output layer $\mathbf{y}(t)$ is computed as:

$$\mathbf{y}(t) = g(\mathbf{V}\mathbf{s}(t)), \quad (15)$$

with the activation functions are

$$g(z_m) = \frac{e^{z_m}}{\Sigma_k e^{z_k}} \quad (16)$$

and the target vector $\mathbf{d}(t)$ represents the phoneme $\mathbf{p}(t+1)$. And

$$\mathbf{e}_h(t-\tau-1) = d_h(\mathbf{e}_h(t-\tau)^T\mathbf{W}, t-\tau-1). \quad (17)$$

In this approach, RNN has been used to learn the probabilities of phoneme sequences of utterances in unsupervised manner. Computational requirements for neural network training are not quite high because of the small size of the phone inventory for a single phone recognizer. The network is trained using back propagation through time (BPTT) algorithm, then the error is propagated through recurrent connections back in time steps t. Hence, the network can remember information for as many time steps as many training examples that were already seen. Moreover, the output layer is factorized into classes to speedup RNN feature reconstruction processes. Every phoneme has been assigned to exactly one class.

3.2. Feature reconstruction based language recognition system

The architecture of the feature reconstruction language recognition system is shown in Fig.2. In VSM part the vector space models are trained by all the training data and classified just like baseline system. In RNN feature reconstruction part, the RNN feature reconstruction processing is described in 3.1. If we employ an SVM as the classifier, the SVM output is:

$$f'(\varphi_{\mathrm{RNN}}(x)) = \sum\nolimits_{l'} \alpha_{l'} K'(\varphi_{\mathrm{RNN}}(x), \varphi_{\mathrm{RNN}}(x_{l'})) + d', \quad (18)$$

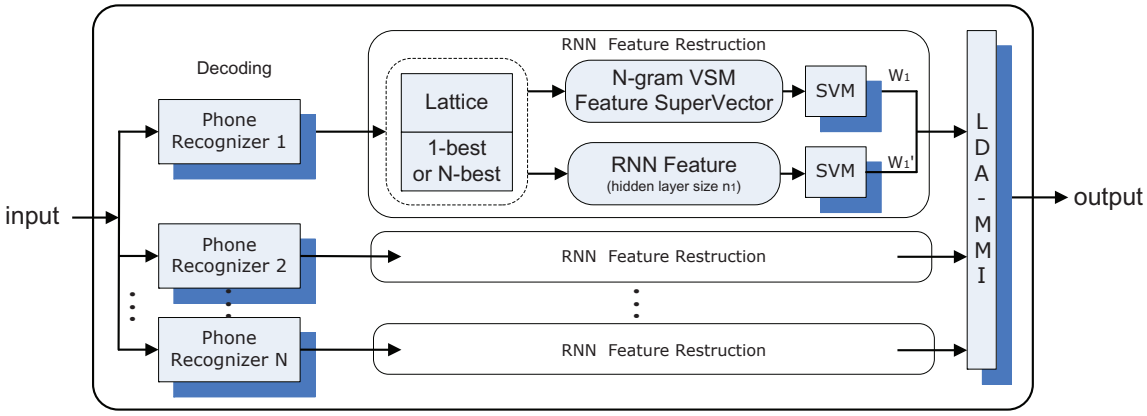

Fig. 2. Architecture of feature reconstruction language recognition system.

where $\varphi_{\mathrm{RNN}}(x_{l'})$ are support vectors obtained from training data. When K' is adopted as TFLLR kernel, K' is computed as following:

$$K'_{\mathrm{TFLLR}}(\varphi_{\mathrm{RNN}}(x_i), \varphi_{\mathrm{RNN}}(x_j)) = \sum_{q=1}^{f*z_h} \frac{\varphi_{\mathrm{r}}(p_q)_{|x_i}, \varphi_{\mathrm{r}}(p_q)_{|x_j}}{\varphi_{\mathrm{r}}(p_q)_{|x_{\mathrm{all}}}}, \quad (19)$$

where $\varphi_{\mathrm{r}}(p_q)_{|x_i}$ denotes the qth element of the vector $\varphi_{\mathrm{RNN}}(x_i)$, and the denominator $\varphi_{\mathrm{r}}(p_q)_{|x_{\mathrm{all}}}$ is the average value of $\varphi_{\mathrm{r}}(p_q)_{|x}$ in all the RNN feature vector used for training. The training is also carried out with a one-versus-rest strategy in feature reconstruction language recognition system.

LDA-MMI method is used to maximize the posterior probabilities of all the belief scores [12] with objective function like this [13]:

$$F_{\mathrm{MMI}}(\lambda) = \sum_{\forall i} \log \frac{p(\mathbf{x}_i|\lambda_{g(i)})P(g(i))}{\sum_{\forall j} p(\mathbf{x}_i|\lambda_j)P(j)}, \quad (20)$$

where

$$\mathbf{x} = [w_1 f(\varphi(x)), w'_1 f'(\varphi_{\mathrm{RNN}}(x)), w_2 f(\varphi(x)), w'_2 f'(\varphi_{\mathrm{RNN}}(x)), ..., w_N f(\varphi(x)), w'_N f'(\varphi_{\mathrm{RNN}}(x))], \quad (21)$$

$g(i)$ denotes its class label. $w, w_1, w_2, ..., w_{\mathrm{N}}, w'_1, w'_2, ..., w'_N$ indicate weights of the belief of the traditional feature and RNN feature. Here $\sum_i \omega_i + \sum_i \omega'_i = 1$. Usually we define $\omega_i = M_i/(\sum_i M_i + \sum_i M'_i)$, $\omega'_i = M'_i/(\sum_i M_i + \sum_i M'_i)$. M'_i is the number of the subset of training utterances that used to produce the feature reconstruction and M_i is the number of the training utterances of phone recognizer i. $P(j)$ is the prior probability of class j. $p(\mathbf{x}|\lambda)$ is weighted Gaussian mixtures.

There are three advantages for feature reconstruction language recognition system. First, RNN feature reconstruction can describe the utterances from a different aspect from the traditional vector space model feature extract method, then the whole system can extract more useful information to classify. Second, the N-grams feature supervector and RNN feature supervector have been shown to be complementary in handling short term contexts and long term contexts, which makes the description of the utterances more precisely and the model more robust. Third, unlike training language model with a high dimensional input and output layer vector as the large vocabulary, the RNN feature reconstruction using a small dimensional input and output layer vector equal as the size of phone inventory of the phone recognizer. Usually the dimension of input and output layer vector is no more than 100, so the RNN feature extraction do not cost much computation. The whole feature reconstruction (FR) system only costs a little more computation than baseline system but gains better performance.

4. EXPERIMENTAL SETUP

4.1. Baseline language recognition system

In this paper a PR-SVM language recognition system is used as baseline system. The first step is to tokenize speech by the means of running Hungarian (HU), Czech (CZ), Russian (RU) Temporal Patterns Neural Network (TRAPs/NN) phone-recognizer that developed by the Brno University of Technology (BUT) [14] and provides the posterior probabilities of the phone occurrences. Then, the decoder named HVite that is produced by HTK [15] is used to produce phone lattices, and a choice of open software (SRILM [16] and rnnlm [17]) is used to produce feature supervector. Then, a popular classifier LIBLINEAR [18] is used to classify. Finally, we use LDA-MMI algorithm [19] for score calibration.

4.2. Test, training and developing dataset

The results in the paper are reported for the test trials of the 2009 National Institute of Standards and Technology Language Recognition Evaluation (NIST-LRE2009). The test data is comprised by 41793 test segments of 23 languages for 30-s, 10-s, and 3-s nominal duration test.

The Call-Home, Call-Friend, OGI, OHSU and VOA Corpus are used in this paper for training.

22701 conversations are selected from the database provided by NIST for the 2003, 2005 and 2007 LRE and VOA as develop database.

4.3. Evaluation measures

In this paper, the performance of language recognition systems is reported in terms of Equal Error Rate (EER) and average cost performance C_{avg} which is defined by NIST LRE 2009 [20].

5. EXPERIMENTAL RESULTS AND DISCUSSION

In this work a PR-SVM [8] language recognition system serves as a baseline system. Here about 1,800,000 utterances are used for training. The BPTT algorithm is used in a block mode and the block size is 10 for at least 5 steps during feature reconstruction. The size of the hidden layer is from 10 to 500. The input layer and the output layer have the same dimension, which is the size of phone inventory for the phone recognizer. Table 1 and Table 2 show the performance of FR system. We use a small subset of training data including about 30,000 utterances to reconstruct N-gram features, HU frontend. Table 1 shows the performance of 1-best phoneme string and Table 2 shows the performance of 50-best phoneme string. Table 1 and Table 2 shows that the performance of FR system is better than baseline but improving slowly with the increasing of the number of hidden layer size. Usually in the training stage the hidden layer size is order of magnitude smaller than input vector, and larger hidden layer does not degrade the performance of the language recognition system but makes the training progress slower. So we can use a small hidden layer size to reduce the computation and get a good performance. And the performance of 1-best is a little better than 50-best because every phoneme string in 50-best merely changes compared with 1-best, so 50-best may bring more noise than information sometimes, but 500-best or more-best can give more information. Actually, Table 1 and Table 2 indicate the short utterances can get the most improvement in FR system. Lacking of phonemes leads to an extremely sparse N-grams feature supervector of the short utterance, which makes the N-grams feature can not describe short utterances precisely. While the RNN feature is a rich representation of short utterances with a fixed dimension.

Table 3 depicts performance of FR system. Fig.3 shows DET curves of both baseline system and FR system on NIST LRE09. Compared with the baseline system, the FR system yielded 1.25%, 3.52% and 14.31% EER, which achieved a 8.76%, 3.82% and 11.93% relative improvements respectively for 30s, 10s and 3s compared to the baseline system.

Table 1. Performance of baseline system and FR system (1-best). NIST LRE 09, HU frontend (EER/Cavg in %). FR-n means the size of the hidden layer of RNN.

	30s	10s	3s
baseline	2.17/1.98	7.61/7.54	23.90/23.42
FR-10	2.19/1.98	7.31/7.23	22.90/22.50
FR-50	2.16/1.99	7.29/7.09	22.24/21.93
FR-500	2.11/2.09	7.17/7.15	22.11/21.69

Table 2. Performance of baseline system and FR system (50-best). NIST LRE 09, HU frontend (EER/Cavg in %). FR-n means the size of the hidden layer of RNN.

	30s	10s	3s
baseline	2.17/1.98	7.61/7.54	23.90/23.42
FR-10	2.14/2.04	7.40/7.22	22.65/22.70
FR-50	2.14/1.99	7.31/7.19	22.37/21.88

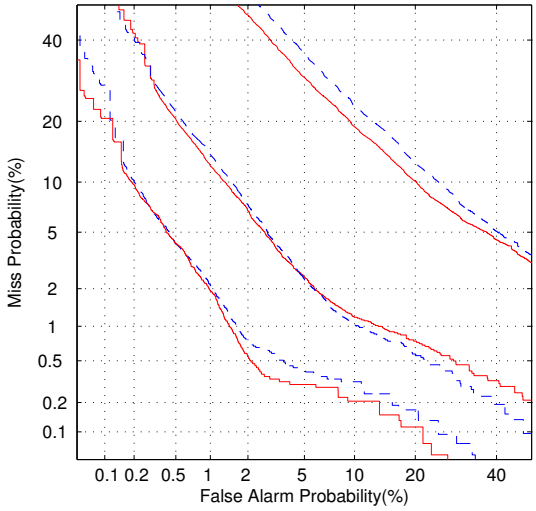

Fig. 3. DET curves of baseline system and FR system for NIST LRE09 (30s, 10s and 3s). The solid line presents baseline system and dashed line presents the FR system.

Table 3. Performance of baseline system and FR system (1-best and 50-best FR, the size of the hidden layer of RNN is 10). NIST LRE 09 (EER/Cavg in %).

	30s	10s	3s
HU(a)	2.17/1.98	7.61/7.54	23.90/23.42
HU-FR(b)	2.06/1.96	7.05/7.04	20.88/20.81
RU(c)	1.90/1.74	5.69/5.47	20.17/20.17
RU-FR(d)	1.79/1.84	5.44/5.42	19.86/19.72
CZ(e)	3.03/2.92	10.07/9.93	25.05/25.62
CZ-FR(f)	2.93/2.85	8.84/8.80	24.00/24.14
(a)+(c)+(e)	1.37/1.35	3.66/3.51	16.25/15.76
(b)+(d)+(f)	1.25/1.17	3.52/3.43	14.31/14.18

6. CONCLUSION

In this paper, an approach to build feature reconstruction language recognition system has been presented. To describe spoken utterances from diversely aspect, the state-of-the-art RNN is employed to reconstruct feature vectors, in which subsets of training data is weighted to produce feature reconstruction for language recognition. The experiments results evaluated on NIST LRE 2009 task show that the relative improvements of the proposed technique are 8.76%, 3.82% and 11.93% for 30s, 10s and 3s over traditional approaches respectively.

7. REFERENCES

[1] Y. K. Muthusamy, E. Barnard, and R. A. Cole, "Reviewing automatic language identification," *Signal Processing Magazine*, vol. 11, no. 4, pp. 33–41, 1994.

[2] M. A. Zissman, "Comparison of four approaches to automatic language identification of telephone speech," *IEEE Transactions on Speech and Audio Processing*, vol. 4, no. 1, pp. 33–44, 1996.

[3] P. A. Torres-Carrasquillo, E. Singer, M. A. Kohler, R. J. Greene, D. A. Reynolds, and J. R. Deller, "Approaches to language identification using Gaussian mixture models and shifted delta cepstral features," *Proc. ICSLP*, pp. 33–36, Sep. 2002.

[4] H. Li, B. Ma, and K.-A. Lee, "Spoken language recognition: from fundamentals to practice," *Proceedings of the IEEE*, vol. 101, no. 5, pp. 1136–1159, 2013.

[5] W.-Q. Zhang, W.-W. Liu, Z.-Y. Li, Y.-Z. Shi, and J. Liu, "Spoken language recognition based on gap-weighted subsequence kernels," *Speech Communication*, vol. 60, pp. 1–12, 2014.

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

[7] T. Mikolov, A. Deoras, S. Kombrink, L. Burget, and J. Cernocky, "Empirical evaluation and combination of advanced language modeling techniques," in *Proc. INTERSPEECH*, 2011, pp. 605–608.

[8] W. M. Campbell, J. P. Campbell, D. A. Reynolds, E. Singer, and P. A. Torres-Carrasquillo, "Support vector machines for speaker and language recognition," *Computer Speech and Language*, vol. 20, no. 2-3, pp. 210–229, Jan 2006.

[9] J. L. Gauvain, A. Messaoudi, and H. Schwenk, "Language recognition using phone latices.," in *Proc. INTERSPEECH*, Jeju Island, Oct 2004, pp. 1283–1286.

[10] W. M. Campbell, J. P. Campbell, D. A. Reynolds, D. A. Jones, and T. R. Leek, "Phonetic speaker recognition with support vector machines," in *Proc. NIPS*, 2003.

[11] T. Mikolov, S. Kombrink, L. Burget, J. H. Cernocky, and S. Khudanpur, "Extensions of recurrent neural network language model," in *Proc. ICASSP*. IEEE, 2011, pp. 5528–5531.

[12] P. Matejka, L. Burget, O. Glembek, P. Schwarz, V. Hubeika, M. Fapso, T. Mikolov, and O. Plchot, "BUT system description for NIST LRE 2007," in *Proc. 2007 NIST Language Recognition Evaluation Workshop*, 2007, pp. 1–5.

[13] D. Povey, *Discriminative training for large vocabulary speech recognition*, Ph.D. thesis, Cambridge, UK: Cambridge University, 2004.

[14] P. Schwarz, *Phoneme recognition based on long temporal context*, Ph.D. thesis, Faculty of Information Technology BUT, 2009.

[15] S. Young, G. Evermann, D. Kershaw, G. Moore, J. Odell, D. Ollason, V. Valtchev, and P. Woodland, *The HTK book*, Entropic Cambridge Research Laboratory Cambridge, 2002.

[16] A. Stolcke et al., "SRILM - An extensible language modeling toolkit.," in *Proc. INTERSPEECH*, 2002.

[17] T. Mikolov, S. Kombrink, A. Deoras, L. Burget, and J. Cernocky, "Rnnlm-recurrent neural network language modeling toolkit," in *Proc. ASRU*, 2011, p. 16.

[18] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin, "Liblinear: A library for large linear classification," *The Journal of Machine Learning Research*, vol. 9, pp. 1871–1874, 2008.

[19] W.-Q. Zhang, T. Hou, and J. Liu, "Discriminative score fusion for language identification," *Chinese Journal of Electronics*, vol. 19, pp. 124–128, Jan 2010.

[20] "The 2009 NIST language recognition evaluation plan," http://www.itl.nist.gov/iad/mig/tests/lang/2009/, Apr 2009.