AUTOMATIC LANGUAGE IDENTIFICATION USING DEEP NEURAL NETWORKS

Ignacio Lopez-Moreno[1], *Javier Gonzalez-Dominguez*[1,2], *Oldrich Plchot*[3], *David Martinez*[4], *Joaquin Gonzalez-Rodriguez*[2], *Pedro Moreno*[1]

[1]Google Inc., New York, USA
[2]ATVS-Biometric Recognition Group, Universidad Autonoma de Madrid, Spain
[3]Brno University of Technology, Czech Republic
[4]Aragon Institute for Engineering Research (I3A), University of Zaragoza, Spain
{elnota,jgd}@google.com

ABSTRACT

This work studies the use of deep neural networks (DNNs) to address automatic language identification (LID). Motivated by their recent success in acoustic modelling, we adapt DNNs to the problem of identifying the language of a given spoken utterance from short-term acoustic features. The proposed approach is compared to state-of-the-art i-vector based acoustic systems on two different datasets: Google 5M LID corpus and NIST LRE 2009. Results show how LID can largely benefit from using DNNs, especially when a large amount of training data is available. We found relative improvements up to 70%, in C_{avg}, over the baseline system.

Index Terms— Automatic Language Identification, i-vectors, DNNs

1. INTRODUCTION

The problem of automatic language identification (LID) can be defined as the process of automatically identifying the language of a given spoken utterance [1]. LID is daily used in several applications such as multilingual translation systems or emergency call routing, where the response time of a fluent native operator might be critical [1] [2].

Even though several high level approaches based on phonotactic and prosody are used as meaningful complementary sources of information [3][4][5], nowadays, many state-of-the-art LID systems still include or rely on acoustic modelling [6][7]. In particular, guided by the advances on speaker verification, the use of i-vector extractors as a front-end followed by diverse classification mechanisms has become the state-of-the-art in acoustic LID systems [8][9].

While previous works on neural networks applied to LID report results using shallow architectures [10][11] or convolutional neural networks [12], in this study, we propose the use of deep neural networks (DNNs) as a new method to perform LID at the acoustic level. Deep neural networks have recently proved to be successful in diverse and challenging machine learning applications, such as acoustic modelling [13] [14], visual object recognition [15] and many others [16]; especially when a large amount of training data is available.

Motivated by those results and also by the discriminative nature of DNNs, which could complement the i-vector generative approach, we adapt DNNs to work at the acoustic frame level to perform LID. Particularly, in this work, we build, explore and experiment with several DNNs configurations and compare the obtained results with several state-of-the-art i-vector based systems trained from exactly the same acoustic features.

To assess the proposed method's performance we experiment on two different and challenging LID datasets: 1. A dataset built from Google data, hereafter, Google 5M LID corpus and 2. The NIST Language Recognition Evaluation (LRE'09). Thus, first, we test the proposed approach in a real application; and second, we check if the same behaviour is observed in a familiar and standard evaluation framework for the LID community. In both cases, we focus on short test utterances (up to 3s).

The rest of this paper is organized as follows. Section 2 presents the i-vector based baseline systems, the proposed DNN architecture as well as the score calibration procedure. The experimental protocol and datasets used are then described in section 3. Results are discussed in section 4. Finally, section 5 is devoted to present conclusions and evaluate proposals for future work.

2. DEVELOPED SYSTEMS

2.1. i-vector Based LID Systems

To establish a baseline framework, we built different state-of-the-art LID acoustic systems based on i-vectors [9]. All those systems, while sharing i-vectors as the same starting point, differ in the type of back-end used to perform the final language classification.

From 39 PLP (13 + Δ + $\Delta\Delta$) feature vectors extracted

978-1-4799-2893-4/14/$31.00 ©2014 IEEE

with a 10ms frame rate over 25ms long windows, we followed the standard recipe described in [17] to obtain i-vectors. We trained a Universal Background Model (UBM) with 1024 components and a 400-dimensional total variability subspace initialized by PCA and refined by 10 iterations of EM. Also, we filtered-out silence frames by using energy-based voice activity detector.

Once the i-vectors for every language were extracted, we used different strategies to perform classification. On the one hand, as a discriminative approach, we performed linear Logistic Regression (LR). On the other hand, two generative approaches were tested, LDA followed by cosine distance (LDA_CD), and a Gaussian modelling to fit the i-vectors of each language, with one (1G) or two components - with and without tied covariances - (2G_TC, 2G). We also explored the effect of using a single shared covariance across the languages (1G_SC) vs. per-language covariances. For further details about this approach, see [9].

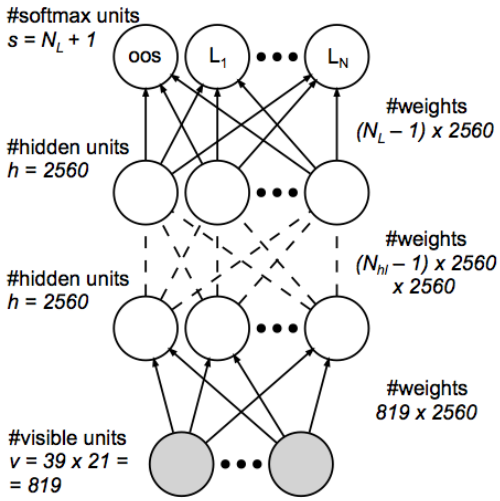

Fig. 1. DNN network topology

2.2. DNN-based LID System

The DNN architecture used in this work is a fully connected feed-forward neural network [18]. The hidden layers contain units with rectified linear activation functions. The output is configured as a softmax layer with a cross-entropy cost function. Each hidden layer contains h (2560) units while the output layer dimension (s) corresponds to the number of target languages (N_L) plus one extra output for the out-of-set (oos) languages.

The DNN works at frame level, using the same features as the baseline systems described above (39 PLP). Specifically, the input layer is fed with 21 frames formed by stacking the current processed frame and its ± 10 left-right context. Therefore, there are 819 (21×39) visible units, v. The number of total weights w, considering N_{hl} hidden layers, can be then easily computed as $w = (v \times h) + ((N_{hl} - 1) \times h \times h) + h \times s$. Figure 1 represents the complete topology of the network.

We trained all the DNN architectures presented in this work using asynchronous stochastic gradient descent within the DistBelief framework [19]. We also fixed the learning rate and minibatch size to 0.001 and 200 samples. Finally, we computed the output scores at utterance level by respectively averaging the log of the softmax output of all its frames (i.e.: log of the predicted posterior probabilities).

2.3. Logistic Regression Calibration

Our scores were calibrated using discriminatively trained, regularized multiclass logistic regression [20]. The calibration was trained in the "cheating" way, that is, using the evaluation scores themselves. The reason, why we performed the cheating calibration, was to concentrate on the ability of the underlying models to discriminate between the given classes. We did not want to introduce other errors coming from over-training the systems on the training data-set and therefore producing miscalibrated scores for our evaluation set.

The L2 regularization penalty weight was chosen prior to training to be proportional to the mean magnitude of the conditioned input vectors (scores) [21].

The calibration uses an affine transform to convert the N_L-dimensional vector of input scores, $\mathbf{s}_t$, for trial t, into a N_L-dimensional calibrated score-vector, $\mathbf{r}_t$

$$\mathbf{r}_t = \mathbf{C}\mathbf{s}_t + \mathbf{d}, \tag{1}$$

The logistic regression parameters are given by $\mathbf{C}$, a full N_L-by-N_L matrix, and $\mathbf{d}$, a N_L-dimensional vector and they are trained by minimizing the multiclass cross-entropy with equalizing the amount of data for individual classes

$$F = \lambda \operatorname{tr}(\mathbf{C}^{\mathrm{T}}\mathbf{C}) - \sum_{i=1}^{N_L} \frac{1}{N_L N_i} \sum_{t \in \mathcal{R}_i} \log \frac{\exp(r_{it})}{\sum_{j=1}^{N}{}_{L} \exp(r_{jt})}, \tag{2}$$

where r_{it} is the ith component of $\mathbf{r}_t$ and $\mathcal{R}_i$ is the set of N_i training examples of language i.

3. EXPERIMENTAL PROTOCOL

3.1. Databases

Google 5M LID Corpus

We generated The Google 5M LID Corpus dataset by randomly picking queries from several Google speech recognition services such as Voice Search or the Speech API.

The Google ASR lattice posteriors were used to discard non-speech queries. Selected queries range from 1s up to 8s nominal duration, with average speech content of 2.1s.

Following the user's phone Voice Search language settings, we labelled a total of ∼5 million utterances, 150k per 34 different locales (25 languages + 9 dialects) yielding ∼87,5h of speech per language and a total of ∼2975h. A held-out test set of 1.5k utterances per language was created while the remainder was used for training and development.

Google queries are not linked to user identity information due to privacy concerns, and therefore, determining the exact number of speakers involved in this corpus is not possible. However, given the selection procedure, it is a reasonable assumption that the number of speakers is very large.

Language Recognition Evaluation 2009 Dataset.

The LRE evaluation in 2009 included, for the first time, data coming from two different audio sources. Besides Conversational Telephone Speech (CTS), used in the previous evaluations, telephone speech from broadcast news was used for both training and test purposes. Broadcast data were obtained via an automatic acquisition system from "Voice of America" news (VOA) where telephone and non-telephone speech is mixed. Up to 2TB of 8KHz raw data containing radio broadcast speech, with the corresponding language and audio source labels were distributed to participants; and a total of 40 languages (23 target and 17 out of set) were included.

Due to the large disparity on training material for every language (from ∼10 to ∼950 hours) and also, for the sake of clarity, we selected 8 representative languages for which at least 200 hours of audio are available: en (US English), es (Spanish), fa (Dari), fr (French), ps (Pashto), ru (Russian), ur (Urdu), zh (Chinese Mandarin). Further, to avoid misleading result interpretation due to the unbalanced mix of CTS and VOA, all the data considered in this dataset belong to VOA.

For evaluation, we used a subset of the official NIST LRE 2009 3s condition evaluation set (as for training, we also discarded CTS test segments), yielding a total of 2916 test segments of the 8 selected languages. That makes a total of 23328 trials.

3.2. Performance Metrics

In order to assess the performance, two different metrics were used. As the main error measure to evaluate the capabilities of one-vs.-all language detection, we use C_{avg} (average cost) as defined in the LRE 2009 [22][23] evaluation plan. C_{avg} is a measure of the cost of taking bad decisions, and therefore it considers not only discrimination, but also the ability of setting optimal thresholds (i. e., calibration). Further, well-known metric Equal Error Rate (EER) is used to show the performance, when considering only scores of each individual language. Detailed information can be found in the LRE'09 evaluation plan [22].

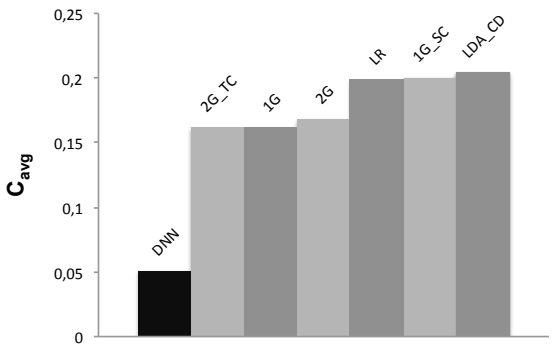

Fig. 2. C_{avg} results on Google 5M LID corpus. 8-hidden layer DNN vs. reference systems based on i-vectors.

4. RESULTS

4.1. Results on Google 5M LID Corpus

As a starting point for this study we compare the performance of the proposed DNN architecture and the reference systems on the large Google 5M LID dataset. Figure 2 shows this comparison. Considering i-vector systems, we found a similar performance for the discriminative back-end, Logistic Regression (LR) and the generative ones, Linear Discriminant Analysis (LDA_CD) and the one based on a single Gaussian with a shared covariance matrix across the languages (1G_SC). Interestingly, increasing to 2 Gaussians and allowing individual covariances matrices (systems 1G, 2G_TC, 2G) a relative improvement of ∼19% is obtained. respect to LR, LDA_CD and 1G systems. This fact suggests that within-class distribution can be different for the individual languages.

Nonetheless, the best performance is achieved by the DNN systems, where the 8-hidden layer DNN proposed architecture yields up to a ∼70% of relative improvement in C_{avg} terms with respect to the best reference system (2G_TC). This result demonstrates the ability of the DNN to exploit discriminative information in large datasets.

4.2. Results on LRE'09

Guided by the results presented above we moved to a more extensive analysis on LRE'09 evaluation data. Per-language results summarized in Table 1, show similar improvements on the LRE'09 dataset. A relative improvement of ∼43% in EER is obtained with the 8-hidden layer DNN, trained with 200h, with respect to the classical i-vector LDA_CD system.

The effect of using different numbers of layers is also highlighted in Table 1, where in addition to the 8-hidden layer DNN and the i-vector LDA_CD system, results with a 2-hidden layer DNN (DNN_2_200h) are also reported. Similarly, although the improvements are more modest than those

	Equal Error Rate (EER in %)								
	en	es	fa	fr	ps	ru	ur	zh	Average
Iv_200h	17.22	10.92	20.03	15.30	19.98	14.87	18.74	10.09	15.89
DNN_2_200h	12.66	5.04	19.67	8.60	17.84	8.75	14.78	5.54	11.61
DNN_8_200h	**8.65**	**3.74**	**17.22**	**7.53**	**16.01**	**5.59**	**13.10**	**4.82**	**9.58**

Table 1. Systems performance (ERR %) per language on LRE'09 (3s test segments)

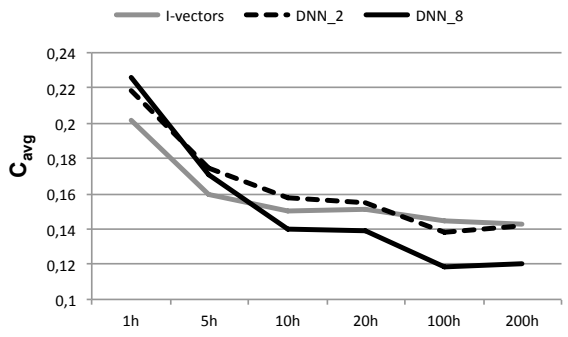

Fig. 3. DNNs vs i-vector performance in function of the per-language number of hours available. Results on LRE'09 dataset.

obtained with the 8-hidden layer network, the DNN_2_200h still outperforms the i-vector based system.

We then explore the effect of having different amounts of training data. Figure 3 shows the i-vector LDA_CD, DNN_2 and DNN_8 systems performance (C_{avg}) as a function of the number of hours per language used for training. With limited amount of data per language ($<$10h), the i-vector system yields the best performance. However, the more hours for training, the higher the improvement of DNN with respect to the i-vector systems. With the greatest amount of data, 200h, the relative improvement of the 8-hidden layer DNN with respect to the i-vector systems is $\sim$15% in C_{avg}.

This behaviour may be for several reasons. With $<$10h per language, the i-vector approach might be favoured by its subspace intrinsic nature. The UBM and the total variability matrix drive modelling to a constrained low-dimensional space. This fact facilitates i-vector approach to quickly retain most important language variations. Also, the number of free parameters to train in each system could play an important role ($\sim$16M, $\sim$9M, $\sim$50M parameters for LDA_CD, DNN_2 and DNN_8 respectively). On the contrary, with abundant data ($>$20h) the i-vector based approach seems to saturate, while DNNs show a high ability to avoid local minima and overfitting even when containing a large number of free parameters (see performance differences between DNN_2 and DNN_8 in Figure 3).

5. DISCUSSION

In this work, we experimented with the use of deep neural networks (DNNs) to automatic language identification (LID). Guided by the success of DNNs for acoustic modelling, we explored their capability to learn discriminative language information from speech signals.

We compared the proposed DNNs architectures to several state-of-the-art acoustic systems based on i-vectors. Results on NIST LRE 2009 (8 languages selected) and Google 5M LID datasets (25 languages + 9 dialects), demonstrate that DNNs outperform, in most of the cases, current state-of-art approaches. This is especially true when large amount of data is available ($>$ 20h), where unlike i-vectors approaches, which seem to saturate, DNNs still learn from data.

On the other hand, DNNs have several drawbacks, including the training time, or the number of parameters to store. Also, adjusting the proper number of hidden layers and units is an empirical exercise for every database. Fortunately, we found that (for the datasets used) moving from 8 to 2 hidden layers, did not have a dramatic impact on performance. Moreover, those adjustments could be done off-line, with testing time still reasonable.

As future work, we will focus on different lines such as establishing a more appropriate averaging of frame posteriors obtained in DNNs, exploring different fusions among DNNs and i-vector systems, or dealing with unbalanced training data.

6. REFERENCES

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

[2] E. Ambikairajah, Haizhou Li, Liang Wang, Bo Yin, and V. Sethu, "Language identification: A tutorial," *Circuits and Systems Magazine, IEEE*, vol. 11, no. 2, pp. 82–108, 2011.

[3] M. Zissman, "Comparison of Four Approaches to Automatic Language Identification of Telephone Speech," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 4, no. 1, pp. 31–44, 1996.

[4] L. Ferrer, N. Scheffer, and E. Shriberg, "A Comparison of Approaches for Modeling Prosodic Features in Speaker Recognition," in *International Conference on Acoustics, Speech, and Signal Processing*, 2010, pp. 4414–4417.

[5] D. Martinez, E. Lleida, A. Ortega, and A. Miguel, "Prosodic features and formant modeling for an ivector-based language recognition system," in *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*, 2013, pp. 6847–6851.

[6] P.A. Torres-Carrasquillo, E. Singer, T. Gleason, Alan McCree, D.A. Reynolds, F. Richardson, and D. Sturim, "The MITLL NIST LRE 2009 Language Recognition System," in *Acoustics Speech and Signal Processing (ICASSP), 2010 IEEE International Conference on*, 2010, pp. 4994–4997.

[7] J. Gonzalez-Dominguez, I. Lopez-Moreno, J. Franco-Pedroso, D. Ramos, D.T. Toledano, and J. Gonzalez-Rodriguez, "Multilevel and Session Variability Compensated Language Recognition: ATVS-UAM Systems at NIST LRE 2009," *Selected Topics in Signal Processing, IEEE Journal of*, vol. 4, no. 6, pp. 1084–1093, 2010.

[8] N. Dehak, P. A. Torres-Carrasquillo, D. A. Reynolds, and Reda Dehak, "Language Recognition via i-vectors and Dimensionality Reduction.," in *INTERSPEECH*. 2011, pp. 857–860, ISCA.

[9] D. Martinez, O. Plchot, L. Burget, Ondrej Glembek, and Pavel Matejka, "Language Recognition in iVectors Space.," in *INTERSPEECH*. 2011, pp. 861–864, ISCA.

[10] R.A. Cole, J.W.T. Inouye, Y.K. Muthusamy, and M. Gopalakrishnan, "Language identification with neural networks: a feasibility study," in *Communications, Computers and Signal Processing, 1989. Conference Proceeding., IEEE Pacific Rim Conference on*, 1989, pp. 525–529.

[11] M. Leena, K. Srinivasa Rao, and B. Yegnanarayana, "Neural network classifiers for language identification using phonotactic and prosodic features," in *Intelligent Sensing and Information Processing, 2005. Proceedings of 2005 International Conference on*, 2005, pp. 404–408.

[12] G. Montavon, "Deep learning for spoken language identification," in *NIPS workshop on Deep Learning for Speech Recognition and Related Applications*, 2009.

[13] A. Mohamed, G.E. Dahl, and G. Hinton, "Acoustic Modeling using Deep Belief Networks," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 20, no. 1, pp. 14–22, 2012.

[14] G. Hinton, Li Deng, Dong Yu, G.E. Dahl, A. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T.N. Sainath, and B. Kingsbury, "Deep Neural Networks for Acoustic Modeling in Speech Recognition: The Shared Views of Four Research Groups," *Signal Processing Magazine, IEEE*, vol. 29, no. 6, pp. 82–97, 2012.

[15] D.C Ciresan, U. Meier, L.M. Gambardella, and J. Schmidhuber, "Deep Big Simple Neural Nets Excel on Handwritten Digit Recognition," *CoRR*, vol. abs/1003.0358, 2010.

[16] D. Yu and L. Deng, "Deep Learning and its Applications to Signal and Information Processing [exploratory dsp]," *Signal Processing Magazine, IEEE*, vol. 28, no. 1, pp. 145–154, 2011.

[17] N Dehak, P Kenny, R. Dehak, P. Dumouchel, and P. Ouellet, "Front-End Factor Analysis for Speaker Verification," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 19, no. 4, pp. 788 – 798, February 2011.

[18] N. Jaitly, P. Nguyen, A. Senior, and V. Vanhoucke, "Application of Pretrained Deep Neural Networks to Large Vocabulary speech recognition," in *Proceedings of Interspeech 2012*, 2012.

[19] J. Dean, G. Corrado, R. Monga, K. Chen, M. Devin, Q. Le, M. Mao, M.A Ranzato, A. Senior, P. Tucker, K. Yang, and A. Ng, "Large Scale Distributed Deep Networks," in *Advances in Neural Information Processing Systems 25*, P. Bartlett, F.C.N. Pereira, C.J.C. Burges, L. Bottou, and K.Q. Weinberger, Eds., pp. 1232–1240. 2012.

[20] C.M. Bishop, *Pattern Recognition and Machine Learning (Information Science and Statistics)*, Springer, 1st ed. 2006. corr. 2nd printing edition, Oct. 2007.

[21] N. Brummer, S. Cumani, O. Glembek, M. Karafiat, P. Matejka, J. Pesan, O. Plchot, M. Soufifar, E. Villiers de, and J. Cernocky, "Description and Analysis of the BRNO276 system for LRE2011," in *Proceedings of Odyssey 2012: The Speaker and Language Recognition Workshop*. 2012, pp. 216–223, International Speech Communication Association.

[22] NIST, "The 2009 NIST SLR Evaluation Plan," www.itl.nist.gov/iad/mig/tests/lre/2009/LRE09_EvalPlan_v6.pdf, 2009.

[23] N. Brummer, *Measuring, Refining and Calibrating Speaker and Language Information Extracted from Speech*, Ph.D. thesis, Department of Electrical and Electronic Engineering, University of Stellenbosch,, 2010.