Using Language Adaptive Deep Neural Networks for Improved Multilingual Speech Recognition

Markus Müller, Alex Waibel

Karlsruhe Institute of Technology
Karlsruhe, Germany
m.mueller@kit.edu

Abstract

Building Large Vocabulary Continuous Speech Recognition (LVCSR) systems for under-resourced languages is a challenging task. While plenty of data is available for English, many other languages suffer from a lack of data. There are different methods for tackling this challenge. One possibility is to use data from different languages to boost the performance of a system for a particular target language. With the emerging of LVCSR systems using neural networks (NNs), many research groups have demonstrated the benefits from using additional data in order to improve the system performance. In this work, we propose a method for providing the language information directly to the network, thus enabling it to become language adaptive. We demonstrate the effectiveness of our approach in a series of experiments.

1. Introduction

With the emergance of Deep Neural Networks (DNNs) in the field of automatic speech recognition, different methods have been explored to improve the performance of Large Vocabulary Continuous Speech Recognition (LVCSR) systems. Although DNNs improve the overall system performance, they require a rather large amount of training data to produce reasonable results.

While there are plenty of resources available for English, this does not necessarily hold true when building a system for another language. One possible solution for this problem is to use data from other languages if there is only a limited amount of data available for a particular target language. Several methods have been explored to make use of multilingual data during system training. By using additional data sets, it is for instance possible to either reduce the training time [1] or decrease the word error rate (WER) [2].

Our proposed method aims towards making better use of the provided multilingual data by explicitly providing a language code to the DNN. By doing so, the DNN becomes aware of the different languages used and is able to implicitly learn language specific features. The resulting DNN is language adaptive (LA-DNN) as it processes the language information in addition to the other input features. We evaluate our proposed method by using different ways of adding the language information to the training pipeline.

This paper is structured as follows: In section 2 we review work related to our experiments. In the following section 3 we describe our proposed method for the network training. Section 4 explains our experimental setup and in section 5 we describe and evaluate the results. Finally, we conclude our paper with section 6 where we review our proposed method and also point towards future work.

2. Related work

Current state-of-the-art speech recognition systems rely on using NNs. The networks are being used in various components like audio pre-processing, language modelling and acoustic modelling. In this work, we concentrate on the use of NNs as a part of the audio pre-processing pipeline and the acoustic model.

2.1. Deep Belief Botteneck Features

Deep Belief Neural Networks (DBNFs)[3] process audio features which were extracted from the raw audio using common approaches like mel-scaled cepstral coefficients (MFCC) or logarithmic mel-scaled spectral coefficients (lMel). DBNFs are feed forward neural networks featuring multiple hidden layers. We first pre-train the network using de-noising auto-encoders [4]. This step initializes the network parameters and guides them into an appropriate range. In the next step, the parameters are fine-tuned using Stochastic Gradient Descent (SGD) [5] with mini-batch updates. For the extraction of the features, the layers after the bottleneck are discarded and the output of the bottleneck layer is used as features.

2.2. Multilingual DBNFs

Since neural networks are good at learning different tasks [6], DBNFs can be trained using multiple languages. Furthermore, [7] has shown that the pre-training step is language independent. Therefore it is possible to increase the performance of the network by using the combined data from multiple languages for training the network. After pre-training, the network is fine-tuned. There are two possibilities to deal

with multiple languages at this stage. It is possible to use a merged phoneme set [8] or share the hidden representations among different languages but use language specific output layers ([9], [10], [11], [12]), see figure 1.

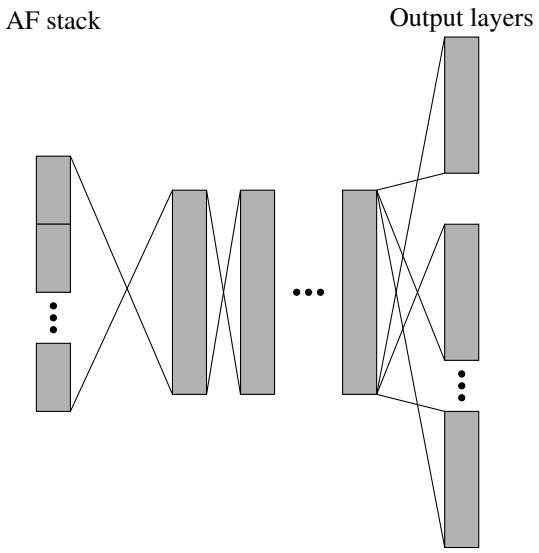

Figure 1: Neural network featuring shared hidden layers and multiple output layers. In our setup, each output layer corresponds to a language specific phone set.

2.3. Augmented Input Features for Neural Networks

Recent publications show that augmenting the input features of the network with additional information like i-Vectors [13] or Bottleneck Speaker Vectors (BSV) [14] increases the overall performance of the system. By providing this additional information, the network adapts to different speakers or acoustic conditions [15]. Since neural networks can process multimodal input data, adding additional information to the features is possible. By doing so, we can provide additional cues to the network. While this was done in the past to provide information about speakers or channels, but, to our knowledge, the use of language codes for building systems in a multilingual environment has not been investigated.

3. Language Adaptive Deep Neural Networks

Augmenting input features with additional information increases the performance of neural networks. Here, we present our approach to add language codes during neural network training in a multilingual environment. By providing this language information in addition to the acoustic features, the network is able take advantage of the language information. As multilingual data can boost the performance of a system when little or no data from the target language is available, we show that this boost can be increased through a language code. As for this code, we chose to encode the language information using 1-of-N coding. This results in a feature vector with one dimension per language.

As pointed out in the related work section, there are multiple possibilities to add data from additional languages throughout the network training. One possibility is to directly merge the available data sets: Create a unified phoneme set, join the different dictionaries and use the audio data jointly to train the system. Another possibility is to build systems for each language individually and then use the individual systems to create language dependent training data for the NNs. It is then possible to share the hidden representations and use language specific output layers. This training technique can be applied to both DBNFs and Hybrid systems. In the latter, the Gaussian Mixture Models (GMMs) are being replaced using a NN.

The language code can be added to the training process of each network. Figure 2 shows the different positions where the language code can be added. It is possible to do an early fusion by appending the language code to the stacked feature frames from the audio pre-processing. Doing so would help the network to discriminate between different languages, but as we will see, this might not be beneficial in all cases. Performing a late fusion is also possible by augmenting the stacked bottleneck features with the language code.

4. Experimental Setup

We conducted a series of experiments in order to assess the performance of our approach. The question is how to augment the existing features with the language code. For training our systems, we use a speech corpus consisting of recordings from Euronews[1], a TV news station [16]. It consists of approximately 70h of acoustic training data per language, sampled at 16 kHz. We use data from 6 languages (English, French, German, Italian, Russian and Turkish), as shown in Table 1. For testing, we used 1.1h of English TV reports.

The pronunciation dictionaries were automatically created using MaryTTS [17]. We selected the languages based on the availability of both recordings from Euronews and pronunciations from MaryTTS. We built the systems using the Janus Recognition Toolkit (JRTk) [18] which features the IBIS decoder [19]. The neural networks were trained using a setup based on Theano ([20], [21]).

Language	Audio Data	# Phonemes
English	72.8h	36
French	68.1h	32
German	73.2h	41
Italian	77.2h	31
Russian	72.2h	27
Turkish	70.4h	31
Total	433.9h	43

Table 1: Overview of used datasets

[1] www.euronews.com

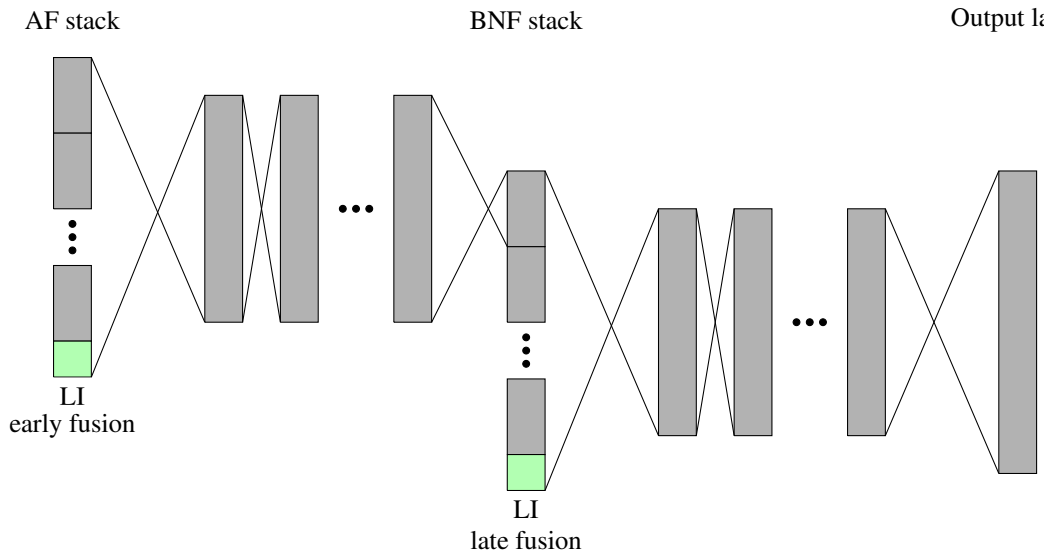

Figure 2: Overview of the network architecture used in our setup. Starting with stacking the acoustic input features (AF), we augment them with a language information (LI) code before feeding them into a DBNF in order to extract BNFs. The BNFs are being stacked as well and the LI code is added. The second DNN computes the phoneme posteriors.

4.1. System Training

For building an initial system, we use a flatstart approach to bootstrap the acoustic models. The audio is pre-processed using 13 dimensional lMEL input features with Δ and $\Delta\Delta$ coefficients which are computed over a window of 16ms that is shifted with 10ms over the audio recording. Based on this inital system, we built a context-dependent system using 6000 context-dependent states. Preliminary experiments have shown that a system using 6000 states has reasonable performance given the amount of available training data.

4.2. DBNF Training

Based on this initial context-dependent system, we extracted samples for training the DBNF network. For training the network, we extracted the samples using a combination of lMel, fundamental frequency variation (FFV) [22] and pitch [23] acoustic features. For the extraction of FFV and pitch, we use a window size of 32ms. The use of additional tonal features has lead to improvements in combination with NNs, even for non-tonal languages such as English [24]. The input features are being stacked using a context of 6 on each side. This results in 13 stacked feature frames being fed into the network at each time step. These stacked frames are then optionally augmented by our 6 dimensional language code which indicates the current language.

The network is layer-wise pre-trained using de-noising auto-encoders. It consists of 5 hidden layers with 1000 neurons per layer. The bottleneck is a narrow layer with only 42 neurons. For fine-tuning, we use stochastic gradient descent with new bob scheduling and log-linear regression. Based on the features extracted by this network, we trained another GMM/HMM system.

4.3. Hybrid System Training

We use the BNF GMM/HMM system to extract a new set of samples for training a DNN. For training this network, we stack features with a context of 7 BNF-frames in each direction, resulting in a total context of 15 frames being fed into the network. This network features 6 hidden layers with a size of 1600 neurons per layer. We use this network as a replacement for the GMMs to estimate the phoneme posterior probabilities. Similar to the training of the DBNF, the input vector for this network is optionally augmented with the language code.

4.4. Merged Phoneme Set

In the first set of experiments, we built a system with language independent models. For training this system, we merged the different training data sets and the pronunciation dictionaries. As we used MaryTTS for generating the dictionaries, we did not need to do a phoneme conversion between the different languages, as all the phonemes already originate from the same phoneme set.

The baseline GMM/HMM system is bootstrapped using all available acoustic data from the 6 languages. This results in 433h of training data for the acoustic model of the system. Based on this initial system, we follow the training procedure described in order to build the BNF based system and the Hybrid system. In order to reduce the training time, we limited the amount of data for the neural network training to 30h per language. To obtain this subset of 30h, we selected a subset of TV reports randomly.

4.5. Language Dependent Phoneme Sets

In a second set of experiments, we built systems with language specific phoneme sets. We used monolingual systems for the extraction of training data for the BNF. Based on this data, we trained a multilingual DBNF by training the hidden layers jointly over all languages and using separate output layers for each language. Based on the multilingual BNF, we again trained systems monolingual for all languages and used them to extract the training data for the Hybrid systems. As for the Hybrid systems, we employed the same training strategy by sharing the hidden layers among languages. The language code was appended to the stacked BNFs.

5. Results

The results section is divided into three different parts. First we present the results from the systems with the merged phoneme set. Next, we show the results from the systems with language specific phoneme sets. This section concludes with a comparison between the multilingual systems and a system trained monolingually.

5.1. Merged Phoneme Sets

The initial GMM/HMM system with a merged phoneme set features a WER of 26.3% as displayed in Table 2. This is rather high, but expected for this type of system: GMM/HMM systems tend to have a poor performance when trained in this multilingual fashion. Using bottleneck features decreases the WER to 21.7% without the language information and 21.2% when adding the language code. The system with the LA-DNN is by 2.4% relative better compared to the system without that additional information. This trend continues for the Hybrid systems. The use of the language information results in a total relative gain of 9.0%. Using a merged phoneme set, adding the language code at both stages (early and late fusion) of network training is beneficial.

System	Baseline	LA-DNN	rel. gain
GMM/HMM	26.3%	26.3%	-
BNF	21.7%	**21.2%**	2.4%
Hybrid	19.3%	**17.7%**	9.0%

Table 2: Overview of results for systems with a merged phoneme set, showing WERs.

5.2. Language Dependent Phoneme Sets

The baseline system with a language dependent phoneme set for English has a WER of 18.9% (see Table 3). This is to a great extend better compared to the system with a merged phoneme set. It is interesting to see that the system with bottleneck features does not benefit from the language code: Providing the language code to the network results in a WER of 18.7%, while the WER is 17.5% when training it without the language information. We therefore use the system without the language code (and the better performance) to write samples for training both Hybrid systems. Based on the performance of the Hybrid systems, it can be seen that adding the language code at the bottleneck layer helps improving the system by 3.5% relative: The system with the language information has a WER of 14.4%, compared to 14.9% WER to the system without. Here, only the late fusion approach leads to improvements.

System	Baseline	LA-DNN	rel. gain
GMM/HMM	18.9%	18.9%	-
BNF	**17.5%**	18.7%	-6.4%
Hybrid	14.9%	**14.4%**	3.5%

Table 3: Overview of results for systems using separate phoneme sets per language, showing WERs.

5.3. Comparison to Monolingual Systems

In a final set of experiments, we compared the performance of monolingual systems to the best multilingual systems. As shown in Table 4, the multilingual systems outperform the systems trained on only one language. Although the relative gain for the hybrid systems (1.4%) decreases compared to the systems using only BNFs (6.3%), we still achieve an improvement by augmenting the input features with the language code.

System	Monol.	ML EN P. Set	rel. gain
GMM/HMM	18.9%	18.9%	-
BNF	18.6%	**17.5%**	6.3%
Hybrid	14.6%	**14.4%**	1.4%

Table 4: Overview of results using language dependent output layers of the neural network, showing WERs.

6. Conclusion

We have presented a method for improving the performance of NN based systems for LVCSR by augmenting the acoustic input features with a language code in a multilingual setup. Gains can be seen throughout different conditions. Depending on the condition, the addition of the language code at either an early and/or a later stage shows the biggest improvements. With the addition of the language information, the DNN becomes language adaptive and is able to better learn the characteristics of different languages.

With our proposed method, the LA-DNN is able to exploit the training data from different languages in a more efficient manner. One of the next steps is to find a replacement for the explicitly coded language information and to auto-

matically extract the language information from the training material in a way similar to i-Vectors or BSVs.

7. References

[1] S. Stüker, M. Müller, Q. B. Nguyen, and A. Waibel, "Training time reduction and performance improvements from multilingual techniques on the babel ASR task," in *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*. IEEE, 2014.

[2] Q. B. Nguyen, J. Gehring, M. Müller, S. Stüker, and A. Waibel, "Multilingual shifting deep bottleneck features for low-resource ASR," in *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*. IEEE, 2014, pp. 5607–5611.

[3] G. E. Hinton, S. Osindero, and Y.-W. Teh, "A fast learning algorithm for deep belief nets," *Neural computation*, vol. 18, no. 7, pp. 1527–1554, 2006.

[4] J. Gehring, Y. Miao, F. Metze, and A. Waibel, "Extracting Deep Bottleneck Features Using Stacked Auto-Encoders," in *Proceedings of the ICASSP*, Vancouver, Canada, May 2013.

[5] L. Bottou, "Stochastic gradient learning in neural networks," *Proceedings of Neuro-Nımes*, vol. 91, no. 8, 1991.

[6] R. Caruana, "Multitask learning," *Machine learning*, vol. 28, no. 1, pp. 41–75, 1997.

[7] P. Swietojanski, A. Ghoshal, and S. Renals, "Unsupervised cross-lingual knowledge transfer in DNN-based LVCSR," in *Proceedings of the Spoken Language Technology Workshop (SLT), 2012 IEEE*, IEEE. IEEE, 2012, pp. 246–251.

[8] N. T. Vu, W. Breiter, F. Metze, and T. Schultz, "Initialization Schemes for Multilayer Perceptron Training and their Impact on ASR Performance using Multilingual Data," in *Proceedings of the INTERSPEECH*, Portland, Oregon, September 2012.

[9] A. Ghoshal, P. Swietojanski, and S. Renals, "Multilingual training of Deep-Neural networks," in *Proceedings of the ICASSP*, Vancouver, Canada, 2013.

[10] S. Scanzio, P. Laface, L. Fissore, R. Gemello, and F. Mana, "On the use of a multilingual neural network front-end," in *Proceedings of the Interspeech*, 2008, pp. 2711–2714.

[11] G. Heigold, V. Vanhoucke, A. Senior, P. Nguyen, M. Ranzato, M. Devin, and J. Dean, "Multilingual Acoustic Models Using Distributed Deep Neural Networks," in *Proceedings of the ICASSP*, Vancouver, Canada, May 2013.

[12] K. Vesely, M. Karafiat, F. Grezl, M. Janda, and E. Egorova, "The language-independent bottleneck features," in *Proceedings of the Spoken Language Technology Workshop (SLT), 2012 IEEE*. IEEE, 2012, pp. 336–341.

[13] G. Saon, H. Soltau, D. Nahamoo, and M. Picheny, "Speaker adaptation of neural network acoustic models using i-vectors," in *Automatic Speech Recognition and Understanding (ASRU), 2013 IEEE Workshop on*. IEEE, 2013, pp. 55–59.

[14] H. Huang and K. C. Sim, "An investigation of augmenting speaker representations to improve speaker normalisation for dnn-based speech recognition," in *Acoustics, Speech and Signal Processing (ICASSP), 2015 IEEE International Conference on*. IEEE, 2015, pp. 4610–4613.

[15] Y. Miao and F. Metze, "Distance-aware dnns for robust speech recognition," in *Sixteenth Annual Conference of the International Speech Communication Association*, 2015.

[16] R. Gretter, "Euronews: a multilingual benchmark for asr and lid," in *Fifteenth Annual Conference of the International Speech Communication Association*, 2014.

[17] M. Schröder and J. Trouvain, "The german text-to-speech synthesis system mary: A tool for research, development and teaching," *International Journal of Speech Technology*, vol. 6, no. 4, pp. 365–377, 2003.

[18] M. Woszczyna, N. Aoki-Waibel, F. D. Buø, N. Coccaro, K. Horiguchi, T. Kemp, A. Lavie, A. McNair, T. Polzin, I. Rogina, C. Rose, T. Schultz, B. Suhm, M. Tomita, and A. Waibel, "Janus 93: Towards spontaneous speech translation," in *International Conference on Acoustics, Speech, and Signal Processing 1994*, Adelaide, Australia, 1994.

[19] H. Soltau, F. Metze, C. Fugen, and A. Waibel, "A one-pass decoder based on polymorphic linguistic context assignment," in *Automatic Speech Recognition and Understanding, 2001. ASRU'01. IEEE Workshop on*. IEEE, 2001, pp. 214–217.

[20] F. Bastien, P. Lamblin, R. Pascanu, J. Bergstra, I. J. Goodfellow, A. Bergeron, N. Bouchard, and Y. Bengio, "Theano: new features and speed improvements," Deep Learning and Unsupervised Feature Learning NIPS 2012 Workshop, 2012.

[21] J. Bergstra, O. Breuleux, F. Bastien, P. Lamblin, R. Pascanu, G. Desjardins, J. Turian, D. Warde-Farley, and Y. Bengio, "Theano: a CPU and GPU math expression compiler," in *Proceedings of the Python for Scientific Computing Conference (SciPy)*, June 2010, oral Presentation.

[22] K. Laskowski, M. Heldner, and J. Edlund, "The Fundamental Frequency Variation Spectrum," in *Proceedings of the 21st Swedish Phonetics Conference (Fonetik 2008)*, Gothenburg, Sweden, June 2008, pp. 29–32.

[23] K. Schubert, "Grundfrequenzverfolgung und deren Anwendung in der Spracherkennung," Master's thesis, Universität Karlsruhe (TH), Germany, 1999, in German.

[24] F. Metze, Z. Sheikh, A. Waibel, J. Gehring, K. Kilgour, Q. B. Nguyen, V. H. Nguyen, *et al.*, "Models of tone for tonal and non-tonal languages," in *Automatic Speech Recognition and Understanding (ASRU), 2013 IEEE Workshop on*. IEEE, 2013, pp. 261–266.