Language Feature Vectors for Resource Constraint Speech Recognition

Markus Müller, Sebastian Stüker, Alex Waibel

Karlsruhe Institute of Technology, 76131 Karlsruhe, Germany
Email: {m.mueller,sebastian.stueker,alexander.waibel}@kit.edu
Web: isl.anthropomatik.kit.edu

Abstract

Deep Neural Networks (DNNs) are a key element of state-of-the-art speech recognition systems. Being a data-driven method, they require a significant amount of training data. There exist scenarios in which such an amount of data is not available for a particular language. Building systems for such resource constrained tasks requires special techniques. One common method is to use data from multiple languages to train the acoustic model.

But there are limitations on knowledge transfer between different languages. By the use of Language Feature Vectors (LFVs), we try to mitigate these limitations by providing language information to DNNs. Similar to i-Vectors for speaker adaptation, LFVs enable DNNs to better capture and adapt to inter language characteristics. Previous experiments have shown that providing LFVs to DNNs improved system performance. In this paper, we show that by adding LFVs the performance gap between mono- and multilingual systems decreases.

1 Introduction

Systems for well resourced languages such as English or German achieve good performance using state-of-the-art methods for system building. In recent years, there has been a rising demand to build systems for lesser resourced languages. Large Vocabulary Continuous Speech Recognition Systems (LVCSR) require a fair amount of training data in order to archive good performance. This makes the task of system building for under resourced languages challenging. But there are different approaches to build systems with only a limited amount of data.

One common technique is to use a mixture of data from different languages in addition to the available data of the target language for acoustic model training. We evaluated various data mixing and training strategies in the past [1]. Training neural networks on data from multiple languages yielded higher recognition accuracy. We recently introduced language adaptive DNNs (LA-DNNs)[2] which captured language specific cues and improved the system performance. We extended our approach by transitioning from an explicit to an implicit adaptation: Instead of modelling the language information directly, we trained a neural network to extract a language feature vector (LFV) that conveys language specific features [3]. We demonstrated that these features carry a richer set of information compared to just the language identity (LID). In addition to that, we could apply this technique also to languages not seen during training.

In this paper we pretend that English is a low resource language. For building a LVCSR we therefore restricted the amount of available data to 30h of transcribed audio. In addition, we also used the task of phoneme boundary detection to evaluate our proposed method. We performed the phoneme boundary detection truly crosslingual, by not using any data from the target language to train our system.

This paper is organized as follows: In Section 2, we provide an overview of related work. In the next section, Section 3, we outline our approach in detail. Section 4 then describes our experimental setup and Section 5 the results. We summarize our findings in the final section in which we also provide an outlook to future work.

2 Related work

State-of-the-art LVCSR systems feature artificial neural networks (ANNs). Neural networks are being used in components like audio pre-processing, language modelling or acoustic modelling. In this paper, we focused on the use of DNNs as part of the audio pre-processing pipeline as well as the acoustic model.

2.1 GMM Based Multilingual Systems

Up until the emergence of neural networks as standard part of LVCSR systems for acoustic modelling, using an approach based on GMM/HMM was common. Bootstrapping the acoustic model multilingually introduces some difficulties. This problem of bootstrapping multi- and cross-lingual GMM/HMM systems has been addressed in the past, e.g., in [4]. Various techniques have been explored to use data from multiple languages for building GMM/HMM based systems [5]. There also exist techniques to bootstrap and build systems cross-lingually [6]. The recognition accuracy of multilingually trained systems is not as good as monolingually trained systems. But there exist scenarios in which it is not feasible to build monolingual systems, e.g., if a system with a universal phoneme inventory is required.

2.2 Multilingual DBNFs

Deep Belief Network Features (DBNFs) benefit from being trained multilingually. Training a DBNF network involves two steps: Pre-training and fine-tuning. The pre-training step is language independent [7]. To fine-tune a network, multiple alternatives exist to make use of data from different languages. One possibility is to share the hidden representations among different languages, but keep the output layers language specific ([8], [9], [10], [11]). Discriminating phonemes in different languages are related tasks. Training DNNs multilingually can therefore be seen as multitask learning [12].

2.3 Feature Augmentation

The use of i-Vectors [13] or Bottleneck Speaker Vectors (BSV) [14] is a common approach to augment the acoustic input features in order to build speaker aware ANNs. Based on i-Vectors, it is also possible to train a speaker adaptive neural network [15]. These different approaches prove that ANNs benefit from additional input features.

This principle can also also be applied to multilingual scenarios. As described in section 3, presenting the lan-

guage information to ANNs leads to improvements in multilingual settings: Either the language identity information [2] alone or high order features encoding a richer set of language properties [3].

2.4 Phoneme Boundary Detection

Documenting unwritten languages is the goal of the BULB project [16]. A first step in documenting unwritten languages is the discovery of the phoneme inventory of these languages. One method to achieve this is to segment recordings into phoneme-like units and then further process these units. To detect phoneme boundaries, it is possible to detect acoustic changes in audio signals[17]. A set of metrics exist to evaluate the detected boundaries [18]. We evaluate the detected boundaries using precision, recall and F-Score.

3 Language Feature Vectors

In the past, we demonstrated that using data from additional languages increased the recognition performance of LVCSR systems [1]. We also showed that augmenting acoustic features with a vector encoding the language identity (LID) further improved the performance [2]. But providing the language information using a one-hot encoding did not reflect the characteristics different languages have. We extended our previous approach by training a DNN to extract LFVs that represent a richer set of language features [3]. This network, if trained on a large enough set of languages, is able to generalize so that it extracts meaningful features even for languages not seen during training.

Initially, this network was built as a combination of two networks: A network for extracting DBNFs and a second network which is trained to discriminate languages. The second stage network was built with DBNFs as input features. To train this network, we used the language identity as targets. The layers of this network featured 1600 neurons. In order to obtain the LFVs, we introduced a bottleneck layer as second last last of the network. It had a size of only 42 neurons. This size is identical to the size of the bottleneck we use to extract DBNFs. This bottleneck forced the network to create a low dimensional representation of features encoding language characteristics. For the extraction of the LFVs, we used the layers up until the bottleneck layer and discarded the other layers.

In this paper, we attempt to simplify this approach by using only one network. We provide the same set of acoustic input features to this network like for the DBNF network. In [3] we determined the optimal context width to span 690ms. In order to cover a window of approximately this size, we fed a context of +/- 30 frames into this network. Each frame is computed over a window of 32ms on the raw audio and this window is shifted with 10ms over the entire recording. We also evaluated additional network hyper parameters. These include the size of the bottleneck layer and the size of the hidden layers. In addition to that, we tried different training strategies by presenting the training data to the network in a different way. Like in [3], the network was trained using data from 9 different languages; Arabic, German, French, Italian, Spanish, Polish, Portuguese, Turkish and Russian.

4 Experimental Setup

We evaluated our proposed method in different scenarios. To conduct our experiments, we used the Janus Recognition Toolkit (JRTk) [19] which features the IBIS single-pass decoder [20]. We trained our neural networks using a setup based on Theano [21].

4.1 Corpora

Similar to [3], we used the Euronews Corpus [22] to train and evaluate our setup. It contained recordings from 10 languages: Arabic, English, French, German, Italian, Polish, Portuguese, Russian, Spanish and Turkish. A detailed overview of available data and number of different recordings is shown in Table 1. Time-aligned transcriptions were provided using a LVCSR system. The provided English test set contained 37 recordings with a total length of 1.2h.

In addition to the Euronews corpus, we also used approximately 1h of transcribed recordings of Basaa. This dataset contained only one speaker and was recorded in a clean environment with 44.1kHz which we downsampled to 16kHz. For details about this dataset, please refer to [23].

Language	Audio Data	# Recordings
Arabic	72.1h	4,342
English	72.8h	4,511
French	68.1h	4,434
German	73.2h	4,436
Italian	77.2h	4,464
Polish	70.8h	4,576
Portuguese	68.3h	4,456
Russian	72.2h	4,418
Spanish	70.5h	4,231
Turkish	70.4h	4,385
Total	715.6h	44,253

Table 1: Overview of used datasets

4.2 Pronunciation Dictionaries

We created the pronunciation dictionaries using the MaryTTS Text-to-Speech engine [24]. For each language, we created pronunciation dictionaries with language specific phoneme sets. In order to build multilingual systems, we merged the different pronunciation dictionaries thereby creating a large multilingual dictionary with a global phoneme set.

4.3 System Training

For bootstrapping our systems, we used a flat start approach where we first built context-independent (CI) systems. Based on these systems, we trained context-dependent (CD) systems. We restricted ourselves to those languages in the Euronews corpus where we could generate pronunciation dictionaries with MaryTTS. These languages are German, English, French, Italian, Turkish, Russian.

4.4 DBNF Training

We used CD systems to label the recordings on frame-level basis for training the DBNF network. Acoustic input features for DBNFs consisted of a combination of lMel, fundamental frequency variation (FFV) [25] and pitch [26].

We extracted lMel with 40 dimensions and tonal features with a dimensionality of 14. This results in a feature vector with 54 dimensions. A previous study has shown that the use of tonal features leads to improvements even if the language is non-tonal [27]. We stacked the input features using a context of +/- 6 frames as input to the neural network. To this stack of 13 frames we added our 42-dimensional LFVs.

The network for DBNF extraction consisted of 6 hidden layers. The second last hidden layer was a bottleneck layer. While the other layers featured 1,000 neurons each, the bottleneck layer was very narrow, having only a size of 42 neurons. The network was pre-trained layer-wise, using de-noising auto-encoders. For fine-tuning, we used stochastic gradient descent with newbob scheduling.

4.5 Hybrid System Training

Based on DBNFs, we re-trained the GMM/HMM systems. Using these second stage systems, we obtained labels to train a second DNN on top of DBNFs. This DNN was used to build a hybrid system. As input features for this system, we used DBNFs stacked with a context of +/- 7 frames. Previous experiments have shown that increasing the context size to 7 for this network leads to improvements. We again appended LFVs to the acoustic input features. The DNN in our hybrid system featured 6 hidden layers with a size of 1600 neurons each. The training procedure was identical to the one for DBNFs.

4.6 LFV Network Training

We considered the problem of language detection related to LVCSR. Therefore, we started with the same network parameters as we would use for speech recognition. In this paper, we would like to evaluate if parts of that setup can be altered to better fit to the task of language recognition. We assumed the language properties to be a static feature over a longer period of time compared to phonemes. In our recent work, we determined a context window length of 690ms to be optimal for this task. In this paper, we evaluated different network hyper parameters. We varied the size of the hidden layers as well as the bottleneck. Although the problem of language detection is related to speech recognition, it might be possible that a different network configurations lead to better results. We therefore omitted the network for the extraction of DBNFs and fed the acoustic input features directly into the LFV network. To cover a context width similar to our existing setup, we increased the size of the context window to +/- 30 frames.

We also evaluated a different training schedule for the NNs. NN training is usually performed doing mini-batch updates. Our standard training setup is based on pfiles[1] for storing the training data. In order for the network to generalize in an optimal way, the training data needs to be shuffled. Our training setup does this by loading the data in fixed chunks into memory and shuffling the data on a per utterance basis in memory. While this works sufficiently well to train NNs for speech recognition, the results when performing language recognition might not be optimal.

We therefore altered our setup by shuffling the entire pfile instead of only shuffling the part of the file that is loaded into memory. By doing so, the network sees a larger variety of speakers during each mini-batch. As evaluation metric for the different network configuration, we used the framewise classification error on the validation set, called validation error.

[1] Feature file archive format created by ICSI

4.7 System Training

For evaluation, we built multilingual systems using data from 6 languages (English, French, German, Italian, Russian and Turkish). Like in [3], we used only a subset of 30h per language of the available training data. This subset was created on a per recording basis. We used a 4-gram language model with a vocabulary size of 100k words for testing. We evaluated our approach using a system with a global phoneme set and pronunciation dictionary. With this single dictionary, we bootstrapped a multilingual system and trained both DBNF GMM/HMM as well as hybrid systems with DBNFs.

For neural network training, we added LFVs to the input features of each network. In case of DBNF hybrid systems, the LFVs were added at two different places: At the DBNF and at the DNN that computes phoneme posteriors. As reference, we included numbers from systems trained without any language information, LID and LFVs. In addition to systems with a merged phoneme set, we also built monolingual systems with multilingual trained neural networks. We bootstrapped CD systems for 6 languages monolingually. The DBNF networks were trained multilingual by sharing the hidden layers but using one output layer per language with language dependent phoneme targets. The same method was used for training the DNN of hybrid systems.

4.8 Cross-lingual Phoneme Boundary Detection

To evaluate our proposed method in a cross-lingual setup, we performed phoneme boundary detection cross-lingually. We measured the accuracy of phoneme boundary detection using data from Basaa. In order to obtain phoneme boundaries as reference, we used a multilingual system and adapted the acoustic models using transcripts of the recordings. We evaluated the F-score of the hypothesized phoneme boundaries with respect to the reference boundaries. For the detection of boundaries, we used a multilingual system to ensure high phoneme coverage. We used this system to recognize phonemes on the test data. While the information about phoneme identities was discarded, we retained and evaluted the detected phoneme boundaries.

5 Results

We first evaluated different hyper parameters for language feature vector extraction. After determining the optimal setup for LFV extraction, we built various systems to evaluate our approach.

5.1 Hyper parameters

We varied the size of the hidden layers as well as the size of the bottleneck. As shown in Table 2, reducing either the size of the hidden layers or the bottleneck layer does not increase the recognition accuracy.

HL size	BN size	Validation error
800	42	0.181
1600	**42**	**0.172**
1600	5	0.178

Table 2: Validation error for different **H**idden **L**ayer and **B**ottl**e**neck configurations for LFV extraction.

5.2 Network architecture

For the next evaluation, we omitted the DBNF network and used an alternative method to randomize the utterances throughout the entire pfile.

Type	Validation error
Baseline	0.172
noDBNF	0.218
noDBNF w/ shuffle	0.204

Table 3: Validation error for different network configurations for LFV extraction.

As seen in Table 3, omitting the DBNF network to preprocess the acoustic input features leads to an increased validation error. The rise in the validation error could be an indication that detecting the language identity is more closely related to phoneme recognition than originally anticipated. Different languages may be identified by distinctive sequences of phonemes. Using DBNFs that encode this phonetic information therefore leads to a better language classification, but using the setup without an DBNF has the advantage of being simpler and requiring less resources. Using an alternate method of data shuffling prior to training leads to a reduction of the validation error. This indicates that the LFV network is more sensitive in terms of speaker variability. In order to better discriminate between languages, it is beneficial to present the network data from a larger set of speakers than just the ones included in the current chunk loaded into memory.

5.3 Multilingual Setup

System	DBNF	Hybrid
w/o LI	21.4%	19.1%
LID	20.7%	17.7%
LFVs	20.7%	16.2%
LFVs w/o DBNF	20.7%	17.7%

Table 4: Overview of results for systems with features. The results are given in WER.

Based on the setup without DBNFs, we see results similar to our setup using only the language identity information. But LFVs even in this configuration without DBNFs still have the advantage of being language independent, i.e. they do not need to be retrained for every new language. For better results, sticking to the approach with two networks is essential.

5.4 Monolingual comparison

In the next set of experiments, we compared the results from the multilingual recognizers having a merged phoneme set with the performance of recognizers with a language dependent phoneme set. The results in Table 5 show the differences between the systems of the two categories. Although the systems with monolingual phoneme sets outperform the multilingually trained systems, the gap in performance decreases by LFVs to 5.8% relative.

System	w/o LI	with LID	with LFV
Monolingual	16.7%	16.6%	15.3%
Multilingual	19.1%	17.7%	16.2%
Loss in perf.	14.4%	6.7%	5.8%

Table 5: WER for language dependent and merged phoneme sets. **with**out **L**anguage **I**nformation, with **L**anguage **Id**entity information and with **L**anguage **F**eature **V**ectors

5.5 Cross-lingual Phoneme Boundary Detection

The final evaluation shows the performance of our approach by detecting phoneme boundaries. We determined the accuracy by computing precision, recall and the F-Score. Table 6 shows the results.

System	Precision	Recall	F-Score
DNN	0.520	0.515	0.518
LFVs	0.542	0.532	0.537
LFVs w/o DBNF	0.537	0.528	0.532

Table 6: Results for cross-lingual phoneme segmentation. Using LFVs leads to improvements.

6 Conclusion and Outlook

The creation of systems in resource constrained environments is a difficult task. But previous works have shown that shortage in training data can be compensated. Multiple techniques for applying data from other languages to increase the performance for a target language exist.

We proposed a method for augmenting the input features of neural networks. Similar to speaker adaptation of neural networks, it is possible to adapt to different languages. Our experiments have shown that providing LFVs to neural networks improve the performance of LVCSR systems in a multilingual environment and clise the gap to monolingual systems further. We also showed that the use of DBNFs is essential.

We applied LFVs to the task of phoneme segmentation. In this scenario, we also saw improvements which proves the universal nature of LFVs.

7 Acknowledgements

This work was realized in the framework of the ANR-DFG project BULB (STU 593/2-1 and ANR-14-CE35-002) and also supported by the French Investissements d'Avenir - Labex EFL program (ANR-10-LABX-0083).

References

[1] M. Müller, S. Stüker, Z. Sheik, F. Metze, and A. Waibel, "Multilingual deep bottle neck features - a study on language selection and training techniques," *Proceedings of the 11th International Workshop on Spoken Language Translation (IWSLT)*, 2014.

[2] M. Müller and A. Waibel, "Using language adaptive deep neural networks for improved multilingual speech recognition," *Proceedings of the 12th International Workshop on Spoken Language Translation (IWSLT)*, 2015.

[3] M. Müller, S. Stüker, and A. Waibel, "Language adaptive dnns for improved low resource speech recognition," in *Proceedings of the Interspeech*, 2016.

[4] T. Schultz and A. Waibel, "Fast bootstrapping of lvcsr systems with multilingual phoneme sets.," in *Eurospeech*, 1997.

[5] T. Schultz and A. Waibel, "Language-independent and language-adaptive acoustic modeling for speech recognition," *Speech Communication*, vol. 35, no. 1, pp. 31–51, 2001.

[6] S. Stüker, *Acoustic modelling for under-resourced languages*. PhD thesis, Karlsruhe, Univ., Diss., 2009, 2009.

[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*, pp. 246–251, IEEE, IEEE, 2012.

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

[9] 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*, pp. 2711–2714, 2008.

[10] 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.

[11] 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*, pp. 336–341, IEEE, 2012.

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

[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*, pp. 55–59, IEEE, 2013.

[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*, pp. 4610–4613, IEEE, 2015.

[15] Y. Miao, H. Zhang, and F. Metze, "Towards speaker adaptive training of deep neural network acoustic models," 2014.

[16] S. Stüker, G. Adda, M. Adda-Decker, O. Ambouroue, L. Besacier, D. Blachon, H. Bonneau-Maynard, P. Godard, F. Hamlaoui, D. Idiatov, G.-N. Kouarata, L. Lamel, E.-M. Makasso, M. Müller, A. Rialland, M. V. de Velde, F. Yvon, and S. Zerbian, "Innovative technologies for under-resourced language documentation:
the bulb project," in *2nd Workshop Collaboration and Computing for Under-Resourced Languages (CCURL 2016)*, 2016.

[17] O. Scharenborg, V. Wan, and M. Ernestus, "Unsupervised speech segmentation: An analysis of the hypothesized phone boundaries," *Acoustical Society of America, Journal of*, vol. 127, no. 2, pp. 1084–1095, 2009.

[18] Y. Qiao, N. Shimomura, and N. Minematsu, "Unsupervised optimal phoneme segmentation: objectives, algorithm and comparisons," in *Acoustics, Speech and Signal Processing (ICASSP), 2008 IEEE International Conference on*, pp. 3989–3992, IEEE, 2008.

[19] 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.

[20] 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*, pp. 214–217, IEEE, 2001.

[21] Theano Development Team, "Theano: A Python framework for fast computation of mathematical expressions," *arXiv e-prints*, vol. abs/1605.02688, May 2016.

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

[23] M. Vetter, M. Müller, F. Hamlaoui, G. Neubig, S. Nakamura, S. Stüker, and A. Waibel, "Unsupervised phoneme segmentation of previously unseen languages," in *Proceedings of the Interspeech*, 2016.

[24] 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.

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

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

[27] 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*, pp. 261–266, IEEE, 2013.