Neural Language Codes for Multilingual Acoustic Models

Markus Müller[1], *Sebastian Stüker*[1], *and Alex Waibel*[1,2]

[1]Karlsruhe Institute of Technology, Karlsruhe, Germany
[2]Carnegie Mellon University, Pittsburgh PA, USA
{m.mueller,sebastian.stueker,alexander.waibel}@kit.edu

Abstract

Multilingual Speech Recognition is one of the most costly AI problems, because each language (7,000+) and even different accents require their own acoustic models to obtain best recognition performance. Even though they all use the same phoneme symbols, each language and accent imposes its own coloring or "twang". Many adaptive approaches have been proposed, but they require further training, additional data and generally are inferior to monolingually trained models. In this paper, we propose a different approach that uses a large multilingual model that is *modulated* by the codes generated by an ancillary network that learns to code useful differences between the "twangs" or human language.

We use Meta-Pi networks [1, 2] to have one network (the language code net) gate the activity of neurons in another (the acoustic model nets). Our results show that during recognition multilingual Meta-Pi networks quickly adapt to the proper language coloring without retraining or new data, and perform better than monolingually trained networks. The model was evaluated by training acoustic modeling nets and modulating language code nets jointly and optimize them for best recognition performance.

Index Terms: speech recognition, connectionist temporal classification, neural networks, language adaptation

1. Introduction

Multilingual speech recognition is a challenging problem, because each of the 7,000 living languages and even dialects or accents require own acoustic model for optimal performance. In low-resource scenarios when only a limited amount of training data from the target language is available, the use of data from additional languages is a well established method to improve the system performance. Training on more data from multiple languages allows the models to generalize better. If enough training data is available, monolingual systems display the best performance and systems trained on a combination of languages may even have an inferior performance. Training on multiple languages introduces additional ambiguity. Corpora from different languages may also be annotated in different manners, e.g. using different phoneme sets. Language adaptation methods are required to improve the performance of such acoustic models.

Language adaptation poses different problems than speaker adaptation. Collecting data from several hundreds of speakers is feasible, while collecting data from the same amount of languages is next to impossible. Training on a small number of languages does not enable the network to generalize across languages. Similar issues for speaker adaptation were reported in the early days of automatic speech recognition (ASR) systems, as these systems were trained on data from an equally low number of speakers.

In this work, we will show how acoustic models can be adapted to languages in a multilingual setting. We have proposed adaptation methods based on language features [3, 4] and now propose a new approach that extends our previous work. We will incorporate two novel aspects: a) the use of adaptive neural language codes (NLCs), which are based on language feature vectors (LFVs) [4], but can be adapted during acoustic model training and b) a network superstructure based on Meta-PI [1, 2], which allows to use pre-trained monolingual subnets in a multilingual system. By combining both approaches, we are able to train a multilingual acoustic models which do not only achieve monolingual performance, also improve beyond monolingual systems.

This paper is organized as follows: In Section 2 we provide an overview of related work in the field. The extraction of neural language codes (NLCs) is outlined in Section 3 and the main network architecture in Section 4. We describe our experimental setup in Section 5 and the results in Section 6. This paper concludes with Section 7, where we also provide an outlook to future work.

2. Related Work

2.1. Multi- and Crosslingual Speech Recognition Systems

Prior to the emergence of neural networks, ASR systems were typically built using a GMM/HMM based approach. Methods for training/adapting such systems cross- and multilingually were proposed to handle data sparsity [5, 6, 7]. The process of clustering context-independent phones into context-dependent ones can also be adapted to account for cross- and multilinguality [8].

2.2. Neural Network Adaptation

Supplying additional features to neural networks for adaptation to certain conditions is a common technique. A very common method for speaker adaptation is using i-Vectors [9, 10] which are a low-dimensional representation of speaker and/or channel characteristics. Based on these vectors, speaker adaptive networks can be trained [11]. These low-dimensional representations can also be extracted by a neural network and are called Bottleneck Speaker Vectors (BSVs) [12].

Based on this idea, we used a similar method for adapting DNNs to multiple languages. First, we utilized only the language identity, encoded via one-hot encoding [13]. This approach was refined by using Language Feature Vectors (LFVs) [4]. Like BSVs, LFVs were extracted using a neural network. In comparison to using the language identity alone, LFVs enable better language adaptation which results in lower WERs.

This work was realized in the framework of the ANR-DFG project BULB (ANR-14-CE35-002).

2.3. RNN Based ASR Systems

Artificial neural network (ANN) based ASR systems have gained a lot of research interest in recent years. Due to increased computing capabilities, more complex architectures can be trained on even more data. A novel approach for building systems is to use recurrent neural networks (RNNs) trained using the connectionist temporal classification (CTC) loss function [14]. Being a powerful tool for sequence classification, RNNs are able to model temporal contexts implicitly. No explicit modelling of context-dependent targets as in traditional systems is required. Phones, graphemes or both can be used as acoustic modeling units [15]. Training on whole words is also possible, given enough training data [16]. Other approaches to train a monolithic acoustic model on multiple languages were also proposed [17].

3. Neural Language Codes

We proposed the use of features encoding language properties [4]. These so-called language feature vectors (LFVs) enable language adaptation of neural networks. The network architecture is shown in Figure 1. To extract those features, we trained a deep neural network (DNN) for language identification. This network featured a bottleneck as second-to-last layer. After training, the layers after this layer were discarded and the output activations of the bottleneck layer were taken as LFVs.

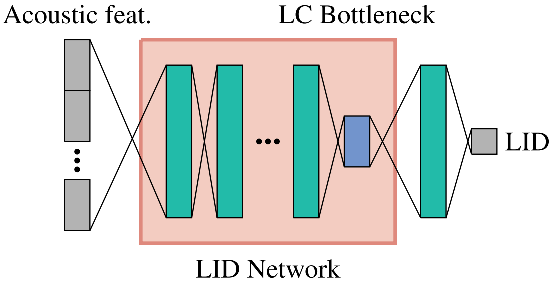

Figure 1: *Language Feature Vectors (LFVs) network architecture*

Adaptation to languages is not as signal related as speaker adaptation. Properties like the length of the vocal tract or the fundamental frequency manifest themselves in the spectrum. Thus, speaker features are typically added at the acoustic feature level. For language adaptation, we showed that adding LFVs deeper into the network [3] does improve the performance, and also optimized this approach [18]. Key is a method called modulation, which is based on Meta-PI networks [1, 2]. These networks feature Meta-PI connections which allow to modulate the output of a neural unit by multiplication with a coefficient. Based on this coefficient, the unit's output value will either be increased or decreased.

As such, in order to modulate the outputs of a whole layer, the number of coefficients has to match the number of outputs. In our experiments we therefore chose the number of units per layer to be a multiple of the dimensionality of LFVs. By stacking LFVs, we could then match the dimensionality of the LFVs to the dimensionality of the layer's outputs. But this also means that the outputs of multiple neurons will be modulated with the same coefficient or the layer's outputs will be divided into multiple groups. The outputs of each unit will be emphasized or attenuated based on the modulation with language codes. The units will in turn become sensitive to language properties. Modulating the outputs in this manner can be considered “intelligent dropout”, as the connections between units are altered in a systematic way, in comparison to randomly dropping entire connections during dropout training [19].

The division of the layer's outputs into groups of equal size for modulation is arbitrary. A better solution would be to break free of this fixed grouping imposed by stacking of the same features multiple times and to furthermore allow updates to the modulation coefficients. We trained a neural language codes (NLC) network to mitigate these issues. The network's architecture is shown in Figure 2. It is trained to generate the stacked LFV output, using both LFVs and acoustic features (multilingual bottleneck features, ML-BNFs) as input. During the training process, it will most likely learn to simply forward the LFVs to the appropriate outputs and to ignore the acoustic features. This network will later be integrated into the network superstructure and adapted based on the new task. By this adaptation, it will learn to transform language feature vectors encoding language properties into neural language codes, which still encode language characteristics, but are now optimized towards speech recognition.

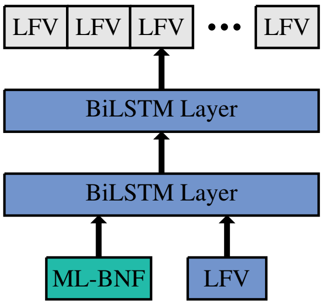

Figure 2: *Neural Language Codes (NLC) network architecture, pre-trained to stack LFVs*

4. Network Architecture

In order to leverage the full potential of monolingual models in our multilingual setup, we explored how monolingual models can be integrated into our network architecture. One method is to use a network architecture based on Meta-PI [1, 2]. The authors presented an approach where parts of the network were trained on different aspects of the same problem. These subnets would then be combined to a larger network superstructure with a trainable component to determine the mixture weights of the outputs from the individual networks.

Based on this setup, we opted for a similar network architecture, where language dependent subnets were pre-trained and then combined to a larger network. On top of these subnets, we would use our default network architecture: A two part BiLSTM network, with the output of the first block being modulated by language features.

4.1. Monolingual Sub Networks

Each monolingual network was trained on a single language only. We trained two networks per language: One using graphemic and one using phonemic targets. We evaluated using networks having both only a quarter or half the amount of LSTM cells per layer in comparison to the main network. We

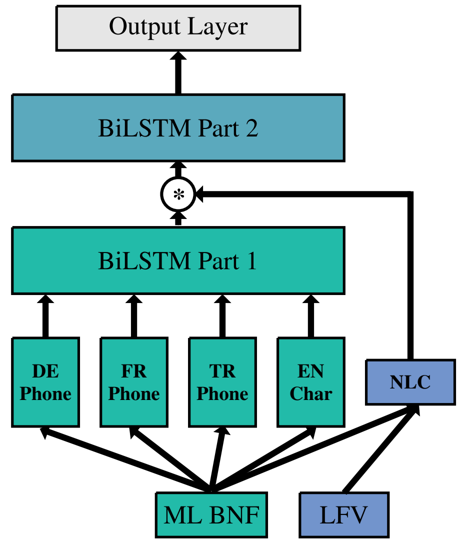

Figure 3: *Network superstructure based on Meta-PI, using Neural Language Codes (NLC) for network modulation.*

also reduced the number of layers from 4 to 3. This network size was selected to limit the number of parameters to not only prevent over-fitting, but also due to memory constraints of the GPUs used for training.

4.2. Main Network

Our main network featured 4 BiLSTM layers with 420 cells each. It was divided into two parts of 2 layers each, with the modulation being applied between these parts. For applying the modulation, we first combined to output of the cells for each direction pairwise by taking the maximum [18]. The combined outputs were then modulated with NLCs.

4.3. Network Training

Initially, the parameters of the pre-trained subnets were loaded. This includes all the language specific nets, as well as the NLC network. The weights of the main network were randomly initialized. The entire network was then jointly trained on the combined graphemic targets of the 4 languages. This joint training allows the individual subnets to adapt to the global task.

5. Experimental Setup

We used data from the Euronews corpus [20], which consists of TV broadcast news recordings from 10 languages. While we used data from 4 languages (English, French, German, Turkish) to train our ASR system, different subsets were used for training the subnets, as indicated in each subsection. The data was filtered removing utterances shorter than 1s or a transcript with more than 639 symbols[1]. This corpus features only very basic annotation of noises with a single noise marker representing any type of noise, e.g. music, human and non-human noises or unintelligible speech. Utterances containing only noise were therefore discarded. In total, 50h of data per language remained after the cleaning process, and this dataset was split into 45h of training and 5h of test data.

As acoustic features, we used multilingual bottleneck features (ML-BNFs). They were trained as part of previous experiments [4] on Euronews data, using a combination of 5 languages (French, German, Italian, Russian, Turkish). The network is fed a combination of logMel and tonal features (FFV [21], pitch [22]), extracted via a 32ms window with a 10ms frame-shift.

5.1. Acoustic Units

Phones and graphemes were used as acoustic modelling units. The pronunciation dictionaries were created automatically using MaryTTS [23]. We mapped the phones of each language using the articulatory features embedded in MaryTTS' language definition files to create a global phone set. In addition, we used a token representing word boundaries.

5.2. Language Dependent Subnets

We trained monolingual networks for each language, using both graphemic and phonetic targets. Each network featured 3 bi-directional layers with either 105 or 210 LSTM cells. The number of cells was chosen based on the size of the main network and we opted for using layers with half or a quarter of LSTM cells. As input features, we used ML-BNFs. A feed-forward layer was used as output layer to map the outputs of the last hidden layer to the output targets. The networks were trained using the CTC loss function, stochastic gradient descent (SGD) and Nesterov momentum [24] with a factor of 0.9. The utterances were sorted ascending by length to stabilize the training, as shorter utterances are easier to align. After training, the output layers were discarded and the outputs of the last hidden layer were used in our network superstructure. By using the features extracted by the last hidden layer instead of the classification result, this setups becomes agnostic towards the set of acoustic units each network uses. As we were using bi-directional LSTM layers, each cell outputs two coefficients, one for each direction. Based on previously reported experiments [18], we opted for taking the pairwise maximum value for each direction, in the notion of maxpool / maxout [25] layers.

5.3. Neural Language Codes

For modulating our main network, we extract NLCs based on a two layer, bi-directional LSTM network with 420 LSTM cells per layer. As input features, we supplied both ML-BNFs and LFVs. The network was trained to output stacked LFVs using mean squared error as loss function. Given the bi-directional nature of the network, we chose to sum the values pairwise for each direction. Applying the sum over the maximum is potentially more stable in approximating real valued outputs. Preliminary experiments showed that no big differences between both methods exist.

5.4. RNN/CTC Network

The main network consisted of 4 layers with 420 BiLSTM cells each. The number of cells was chosen to be a multiple of the dimensionality of the LFVs. The network was split into two parts with 2 layers each. The first part used the combined outputs of the language dependent subnets as input features. The outputs were merged pairwise using the same maxpool strategy as the subnets. The modulation is then applied prior to feeding the outputs into the second part. A feed-forward layer then mapped the outputs of the last LSTM layer to the targets.

[1] Internal limitation within the implementation of CUDA/warp-ctc, see: https://github.com/baidu-research/warp-ctc, accessed 2018-03-16

5.5. Grapheme Based RNN LM

For decoding, we used a character based RNN LM. It was trained as described in [26] on the transcripts of the training utterances (110k sentences). We first trained a baseline LM with a single layer of 1024 BiLSTM units. We later re-fined it ("new LM" in Table 2) by optimizing the number of BiLSTM cells. In a series of experiments we determined 512 BiLSTM cells to be the optimal number, resulting in the lowest WER.

5.6. Training Strategy

The network superstructure was trained in multiple steps. First, the language dependent subnets were trained individually, so was the NLC network. These networks were then combined with the main network. We evaluated using different combinations of source nets, mixing nets trained on graphemic and phonemic targets (as shown in Figure 3). The entire architecture was trained jointly, allowing updates to the parameters of all networks. Similar to training the subnets, SGD and Nesterov momentum [24] with a factor of 0.9 were used for training. But given the increased parameter count, we also applied Dropout training with a factor of 0.2 to prevent over fitting.

5.7. Evaluation

The network performance was evaluated using both character error rate (CER) and word error rate (WER). For decoding, we used the same procedure as in [14] and greedily searched for the best path. WER results were obtained by performing a decoding using the RNN LM. We compared training multilingual networks for various conditions. As contrasting experiment, we trained an English monolingual system.

6. Results

We first trained monolingual subnets on graphemes and phonemes using different layer sizes and evaluated their performance. Next, we combined these language dependent subnets in our network superstructure.

6.1. Monolingual Subnets

The CERs of our individual monolingual subnets are shown in Table 1. Training networks with only 105 BiLSTM cells per layer results in higher CERs compared to using 210 cells. A network of this size is too small for modelling the acoustics entirely, but we are using it only as part of our superstructure to extract language dependent features. On the other hand, having source nets with too many parameters may render the superstructure prone to over fitting. Comparing the CERs of nets trained on graphemes and phonemes, the error rate of the German and Turkish grapheme based setup is lower in comparison to their phoneme based counterpart. Potential reasons are a) the pronunciation dictionaries may be of varying quality as they were generated completely automatic and b) Turkish as well as German have easier pronunciation rules than English or French.

Type	Size	DE	EN	FR	TR
Phone	105	9.0	14.3	11.7	7.0
Phone	210	7.2	12.2	8.6	5.9
Grapheme	105	8.3	16.5	13.3	7.2
Grapheme	210	6.4	13.1	9.6	5.6

Table 1: *CER of monolingual subnets*

6.2. Multilingual Systems

We evaluated our multilingual setup using English data, with WERs shown in Table 2. As baseline experiment, we trained a system using English data only (setup 1). Training the same setup jointly on data from multiple languages (setup 2) increases the WER on English. Applying language codes via modulation (setup 3, [3]) improves the WER to 26.3%. Pretraining the networks using phonetic information from other but not the target language further reduces to WER to 25.4% (setup 4, [18]), which almost achieves parity with the monolingual baseline (setup 1). Switching then to our network superstructure based on Meta-PI and applying adaptive neural language codes (setup 5), the WER decreases further. This system not only achieves parity with the monolingual baseline, but even surpasses it.

On the best results, we then applied an optimized language model, which reduced the WERs of both systems (6 and 7), with the adapted system (7) again outperforming the monolingual setup (6).

Setup	WER
1) Monolingual baseline	25.3%
2) No adaptation	27.4%
3) LFV Modulation	26.3%
4) Phonetic pre-training	25.4%
5) Meta-PI + NLC	**24.2%**
6) Monolingual baseline (new LM)	24.2%
7) **Meta-PI + NLC (new LM)**	**23.5%**

Table 2: *WER of network superstructure, evaluated on English*

7. Conclusion

We presented a language adaptation method for multilingual RNN/CTC based speech recognition systems. It enables systems with a multilingual acoustic model to not only achieve parity with monolingual systems, but also to improve beyond the monolingual baseline. Training acoustic models on data from multiple languages is a challenging problem because the acoustic modelling units between languages may differ. This is especially problematic if graphemes are used as acoustic modelling units. But by applying neural adaptation techniques proposed here, we are able to mitigate these issues and enable the adaptation of the acoustic model to multiple languages with not only achieving the same performance than the monolingual baseline, but also exceeding it.

Future work includes the integration of more languages into the setup, which we expect will allow for better generalization across languages. But also the use of other acoustic units like, e.g., BPE (byte pair encoding) units should provide better results. We also investigate the use of the proposed adaptation method to new domains.

8. References

[1] J. B. Hampshire and A. H. Waibel, "The Meta-Pi network: Connectionist rapid adaptation for high-performance multi-speaker phoneme recognition," in *Acoustics, Speech, and Signal Processing, 1990. ICASSP-90., 1990 International Conference on*. IEEE, 1990, pp. 165–168.

[2] J. B. Hampshire and A. Waibel, "The Meta-Pi network: Building distributed knowledge representations for robust multisource pattern recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 14, no. 7, pp. 751–769, 1992.

[3] M. Müller, S. Stüker, and A. Waibel, "Multilingual adaptation of RNN based ASR systems," in *Acoustics, Speech and Signal Processing (ICASSP), 2018 IEEE International Conference on*. IEEE, 2018.

[4] ——, "Language Adaptive DNNs for Improved Low Resource Speech Recognition," in *Interspeech*, 2016.

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

[6] ——, "Multilingual and crosslingual speech recognition," in *Proc. DARPA Workshop on Broadcast News Transcription and Understanding*. Citeseer, 1998, pp. 259–262.

[7] S. Stüker, "Acoustic modelling for under-resourced languages," Ph.D. dissertation, Karlsruhe, Univ., Diss., 2009, 2009.

[8] ——, "Modified polyphone decision tree specialization for porting multilingual grapheme based ASR systems to new languages," in *Proceedings of the 2008 IEEE International Conference on Acoustics, Speech, and Signal Processing*. Las Vegas, NV, USA: IEEE, April 2008, pp. 4249–4252.

[9] N. Dehak, P. J. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet, "Front-end factor analysis for speaker verification," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, no. 4, pp. 788–798, 2011.

[10] G. Saon, H. Soltau, D. Nahamoo, and M. Picheny, "Speaker adaptation of neural network acoustic models using i-Vectors," in *ASRU*. IEEE, 2013, pp. 55–59.

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

[12] H. Huang and K. C. Sim, "An investigation of augmenting speaker representations to improve speaker normalisation for DNN-based speech recognition," in *ICASSP*. IEEE, 2015, pp. 4610–4613.

[13] M. Müller and A. Waibel, "Using language adaptive deep neural networks for improved multilingual speech recognition," *IWSLT*, 2015.

[14] A. Graves, S. Fernández, F. Gomez, and J. Schmidhuber, "Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks," in *Proceedings of the 23rd international conference on Machine learning*. ACM, 2006, pp. 369–376.

[15] D. Chen, B. Mak, C.-C. Leung, and S. Sivadas, "Joint acoustic modeling of triphones and trigraphemes by multi-task learning deep neural networks for low-resource speech recognition," in *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*. IEEE, 2014, pp. 5592–5596.

[16] H. Soltau, H. Liao, and H. Sak, "Neural speech recognizer: Acoustic-to-Word LSTM model for large vocabulary speech recognition," *arXiv preprint arXiv:1610.09975*, 2016.

[17] S. Watanabe, T. Hori, and J. R. Hershey, "Language independent end-to-end architecture for joint language identification and speech recognition," in *Automatic Speech Recognition and Understanding Workshop (ASRU), 2017 IEEE*. IEEE, 2017, pp. 265–271.

[18] M. Müller, S. Stüker, and A. Waibel, "Enhancing multilingual graphemic RNN based ASR systems using phone information," *Elektronische Sprachsignalverarbeitung (ESSV) 2018*, 2018.

[19] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *The Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, 2014.

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

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

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

[23] 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.

[24] I. Sutskever, J. Martens, G. Dahl, and G. Hinton, "On the importance of initialization and momentum in deep learning," in *Proceedings of the 30th International Conference on Machine Learning (ICML-13)*, 2013, pp. 1139–1147.

[25] I. J. Goodfellow, D. Warde-Farley, M. Mirza, A. Courville, and Y. Bengio, "Maxout networks," *arXiv preprint arXiv:1302.4389*, 2013.

[26] T. Zenkel, R. Sanabria, F. Metze, J. Niehues, M. Sperber, S. Stüker, and A. Waibel, "Comparison of decoding strategies for CTC acoustic models," *arXiv preprint arXiv:1708.04469*, 2017.