MULTILINGUAL SHIFTING DEEP BOTTLENECK FEATURES FOR LOW-RESOURCE ASR

Quoc Bao Nguyen, Jonas Gehring, Markus Müller, Sebastian Stüker and Alex Waibel

Institute for Anthropomatics, Karlsruhe Institute of Technology, Germany

ABSTRACT

In this work, we propose a deep bottleneck feature architecture that is able to leverage data from multiple languages. We also show that tonal features are helpful for non-tonal languages. Evaluations are performed on a low-resource conversational telephone speech transcription task in Bengali, while additional data for DBNF training is provided in Assamese, Pashto, Tagalog, Turkish, and Vietnamese. We obtain relative reductions of up to 17.3% and 9.4% WER over mono-lingual GMMs and DBNFs, respectively.

Index Terms— Deep Neural Networks, Multilingual Deep bottleneck features, Low-Resource ASR

1. INTRODUCTION

The models of a classical automatic speech recognition system are usually trained on data from one language only. As a consequence, the resulting recognition system is only able to recognize speech from that language. Further, large amounts of training data from the target language of the recognition system need to be available in order to estimate the model parameters robustly. In multilingual speech recognition systems, the models of the system (most prominently the acoustic model) are trained on data from multiple languages [1]. This approach has two advantages. First, the resulting model is in principle capable of recognizing speech coming from any of the languages present in the training data. Second, research has shown that multilingual acoustic models are well suited for initializing acoustic models for new languages, and reducing the amount of training material needed for a new, previously unseen target language [1]. The use of multilingual models in acoustic modeling is especially of use when only small amounts of training data in the target language are available, and the available time for training the new model is limited. Multilingual models can be trained in advance before the need of a recognition system in a new language arises, and can then reduce the training time in that new language.

In the IARPA sponsored Babel[1] program, we face exactly this challenge. Here, the task is to create keyword search systems in new languages with only 10 hours or less of available training data. Also, towards the end of the project, the training time allowed for creating a new system will be reduced to one week. Modern keyword search systems often make use of the result of a large vocabulary continuous speech recognition (LVCSR) system for performing the task. Therefore, in Babel we need to be able to build speech recognition systems for new languages with very little training data in a very short time frame.

Recently, the use of multilingual modeling techniques has been extended from the acoustic model to the pre-processing component of a speech recognition system. With the advent of deep bottleneck features (DBNFs) which make use of deep neural networks (DNNs) the feature extraction aspect of a recognition system now also contains a component in need of training, normally on data from the target language of the recognition system [2]. But multilingual modeling techniques can also be applied here, and have been shown to produce competitive results [3].

In this paper we extend the notion of multilingual modelling to a new architecture of DBNFs, which we call shifting DBNFs (SDBNFs) [4]. We show how training *shifting deep bottleneck features* (SDBNFs) on multiple languages can lead to better performance than training on only the target language of the recognition system, especially in situations with sparse data. For this we compare the performance of different SBNFs and classical standard DBNFs on Bengali as target language, and improve their performance by adding data from the languages Assamese, Pashto, Tagalog, Turkish, and Vietnamese.

2. RELATED WORK

Previous work has shown that neural networks offer the ability to train shared hidden representations across different

The authors would like to thank Joshua Winebarger for his assistance in proofreading this paper.

Supported in part by the Intelligence Advanced Research Projects Activity (IARPA) via Department of Defense U.S. Army Research Laboratory (DoD/ARL) contract number W911NF-12-C-0015. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright annotation thereon. Disclaimer: The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of IARPA, DoD/ARL, or the U.S. Government. This effort uses the IARPA Babel Program language collection releases IARPA-babel{102b-v0.4,103b-v0.3,104b-v0.4bY,105b-v0.4,106-v0.2f,107b-v0.7}.

[1] http://www.iarpa.gov/Programs/ia/Babel/babel.html

tasks. This works particularly well for speech recognition, where different languages have distinct sounds, but may also share acoustic cues which can be learned simultaneously on multiple languages.

Consequently, several recent works on training multilingual DBNFs have been published. Successful demonstrations include the training of feature extraction networks in which all layers are shared and the output layer either predicts the members of a merged phone set [5] or contains a language-specific layer [6] [7] [8] [9].

Since most modern DNN acoustic models are pre-trained in an unsupervised fashion, it is also possible to use multiple languages during pre-training only, and [10] has shown that pre-training is indeed language-independent. Furthermore, Gehring et al. proposed several multilingual deep neural network architectures with the connection of bottleneck feature extraction and acoustic model networks in order to create significantly better acoustic models for a low-resource target language [4].

In this paper, we focus on multilingual bottleneck features with shared hidden representations and language specific-output layers by presenting a new type of DBNFs, SDBNFs. We also add tonal features to the input features of our SDBNF network in order to investigate their effect on a multilingual setup containing non-tonal languages.

3. PRE-PROCESSING PRIOR TO THE DBNF NETWORK

In the past we have experimented with different kinds of inputs to DBNF networks, such as mel-scaled cepstral coefficients (MFCCs), logarithmic mel-scaled spectral coefficients and minimum variance distortionless response (MVDR) coefficients [11]. Also, we have examined the use of features targeting the tonal part of such languages as Vietnamese, and how their use affects the recognition performance when applied to non-tonal languages. We have shown in [12] that stacking MFCC, MVDR and two types of tonal features give the best performance, relative to any of these features alone.

3.1. MFCC and MVDR Features

We extract the features for the MFCCs by using a window of 32ms in length and a window shift of 10ms for short time spectral analysis. Our Mel filterbank extracts 30 coefficients. Instead of using an inverse discrete Fourier transform we use a discrete cosine transformation for the transition into the cepstral domain, where we reduce the number of coefficients to 20 by liftering.

In addition to MFCC features, we also applied an MVDR spectrum, to see how much combining multiple features helps on its own. In this work, we use twice-warped MVDR [13]. Stacking MFCCs and MVDRs at the input of a DNN was found to be helpful in experiments that were part of the NIST 2013 OpenKWS evaluation [2]. While MFCC and MVDR features are fundamentally similar and equally powerful, they are nonetheless complementary. Training a system using both gives gains simply by increasing the robustness of the extraction.

Fundamentally different from spectral features, which capture the envelope of the speech signal, "pitch" features capture variations in the fundamental frequency of the speaker's voice and are typically used in addition to spectral features.

3.2. Fundamental Frequency Variation (FFV) Features

FFV [14] features have previously been used in tasks such as speaker verification. When compared to "standard" pitch-based features, their main advantage is that no explicit segmentation into speech and silence segments (for which pitch is not defined) is necessary.

Rather than locating the peak in the FFV "spectrum" (which is defined over $\tau \in [-\infty, \infty]$), we apply a filterbank, which attempts to capture meaningful prosodic variation, and contains a trapezoidal filter for perceptually "flat" pitch, two trapezoidal filters for "slowly changing" (rising and falling) pitch, and two additional trapezoidal filters for "rapidly changing" pitch. In addition, the filterbank contains two rectangular extremity filters, as unvoiced frames have flat rather than decaying tails. This filterbank reduces the input space to 7 scalars per frame, which we use as additional "FFV" features in the final input vector.

3.3. Pitch Features

In this work, we extract pitch features using the approach described in [15]. We compute a Cepstrogram with a window length of 32 msec, and use dynamic programming to find the best path over time for the location of the maximum in these coefficients under certain constraints, like maximum pitch change per time unit. Additionally, we compute delta and double delta features using the three left and right neighbors as well as frame-based cross-correlation. This results in 8 additional coefficients (1 pitch, 6 delta and double-delta features and cross-correlation). These 8 coefficients are added to the original MFCC+MVDR feature vector.

4. MULTILINGUAL SHIFTING DEEP BOTTLENECK FEATURES

4.1. Deep Bottleneck features

Standard bottleneck features as first described by Grezl et al. are a common part of many automatic speech recognition setups [16]. In this system, a feed-forward neural network is trained as a discriminative feature extractor that predicts, e.g.,

[2] http://www.nist.gov/itl/iad/mig/openkws13.cfm

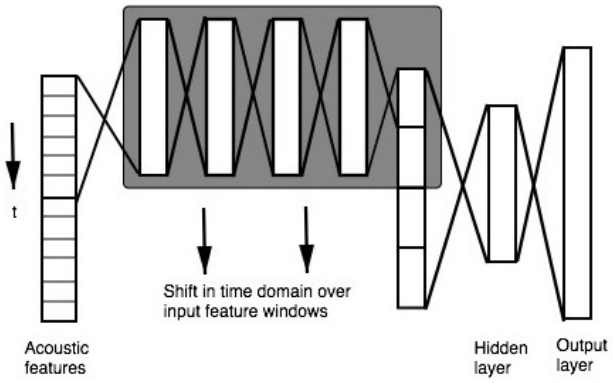

Fig. 1. Proposed Shifting Deep Bottleneck features Architecture

polyphone states from windows of standard speech recognition features like mel-frequency cepstral coefficients. This network contains a narrow "bottleneck" hidden layer with only a few units, which is placed between two larger layers. By computing the activation of units in the bottleneck layer, the network performs a non-linear discriminative dimensionality reduction of its original input. It can then be shown that the bottleneck units provide features for GMM/HMM setups that result in superior recognition accuracy. In previous works, deep learning techniques were applied in BNF training, and they found that pre-training of the individual layers as restricted Boltzmann machines (RBMs) [17] or with a stack of auto-encoder layers [2] were a crucial part in obtaining good features.

4.2. Shifting Deep Bottleneck features

In the pre-processing stage of our speech recognizers, we extract stacked DBNF features before applying LDA. However, typically the training of the DBNF does not employ a stacked approach. To eliminate this mismatch, we conducted experiments wherein the DBNF architecture is extended by training the feature extraction part of the DBNF (that which is used later for preprocessing) on adjacent input feature windows. First, we train a classical DBNF network as described in the preceding section. Thereafter, we refine the network by training it in the following way. In the forward pass of network training, we compute the bottleneck layers from adjacent frames and stack them. Then in backward propagation, we average the gradients for the bottleneck layer. The hidden layer and the output layer are thereby connected to a stack of DBNFs trained on these adjacent windows. This approach is similar to the TDNNs proposed in [18] or later convolutive networks in [19], where the parameters of time shifted copies are shared and scaled. A difference however between our approach and theirs is that instead of simply training new bottleneck features in the shifting process, we refine existing DBNFs trained in the first pass.

Corpus	Language	Abbre	Size
IARPA-babel103b-v0.3	Bengali	BEN	10 h
IARPA-babel102b-v0.4	Assamese	ASM	53 h
IARPA-babel104b-v0.4bY	Pashto	PUS	79 h
IARPA-babel106-v0.2f	Tagalog	TGL	73 h
IARPA-babel105b-v0.4	Turkish	TUR	72 h
IARPA-babel107b-v0.7	Vietnamese	VIE	79 h

Table 1. *Corpora used for multi-lingual network training.*

4.3. Multilingual Deep Bottleneck Features

In order to extract robust DBNF features from multilingual resources, we focus on training bottleneck feature extracting networks using both baseline and shifting DBNF architectures. The networks use shared hidden presentations and language-specific output layers, which avoid the mapping of phonemes of different languages to a common set as in [6] [7]. The auto-encoders used to initialize the hidden layers prior to the bottleneck are pretrained on multiple languages.

5. EXPERIMENTS SETUP AND RESULTS

5.1. Copora and Baseline Description

We performed experiments with various corpora listed in Table 1. All corpora contain narrow-band, conversational telephone speech from land lines as well as mobile phones. For the training of the target language, only 10 hours of data were provided. And additional 356 hours of data from other languages was available for our use.

As a baseline, we used a system which was trained using a multilingual bootstrap (MLBootstrap) technique. For preprocessing, we used a standard MFCC front-end. For bootstrapping, we took the already trained models from four languages (Cantonese, Turkish, Vietnamese and Pashto) to estimate the initial model parameters for Bengali, our target language. This was possible as all data from the BABEL project uses X-SAMPA as a common phoneme set. Based on these initial models, we built first a context-independent system and then a context-dependent system with 2000 models. We trained our networks to predict roughly 2000 context dependent targets from different features: (1) the combination of 20 MFCC and 20 wMVDR coefficients(MFCC+MVDR) plus tone (7 FFVs and 8 Pitch) features and (2) the combination of 20 MFCC, 20 wMVDR coefficients without tone. These were extracted from 32 ms windows with a 10 ms frame shift.

Hidden layers for DBNF networks were pre-trained in an unsupervised manner as denoising auto-encoders, in which

a single layer is trained to properly reconstruct its input after random corruption has been applied [20]. The input vector was deliberately corrupted by applying standard masking noise to set 20% of their elements randomly to zero. For supervised fine-tuning, we used the newbob algorithm in which two separate thresholds control the application of learning rate decay and the total duration of training by monitoring frame-level classification accuracy on a held-out validation set. The DBNF networks contain four auto-encoder layers with 1200 units each, i.e. seven layers in total (with bottleneck, additional hidden layer and output layer). 42 units were used in the bottleneck layer, whereas the layer afterwards contained 1200 units. A 3-gram language model was built from the reference transcriptions of the Bengali corpus. Decoding was done with the JANUS speech recognition toolkit [21] using networks previously trained on GPUs with Theano[3].

5.2. Results

Table 2 lists the performance of baseline systems on the Bengali target language in terms of the word error rate (WER). The GMM system is a context-dependent system using the same states as the DBNFs setups and was trained from the same alignment described in the previous section. The GMM baseline trained with MLBootstrap performs 1% absolute better than the GMM baseline from flatstart. A standard DBNF system does not provide much improvement in this low-resource condition (about 7.6% and 8.7% relative on MFCC+MVDR and MFCC+MVDR+tone respectively). Applying tonal features (FFV+Pitch) together with the combination of MFCC and MVDR features results in systems which also outperform the non-tonal DBNF systems by 1% absolute.

Systems	**Features**	**WER(%)**
Baseline flatstart	MFCC13	79.1
Baseline MLBootstrap	MFCC13	78.0
DBNFs flatstart	MFCC+MVDR	74.7
DBNFs MLBootstrap	MFCC+MVDR	72.1
SDBNFs MLBootstrap	MFCC+MVDR	71.8
DBNFs MLBootstrap	MFCC+MVDR+tone	71.2
SDBNFs MLBootstrap	MFCC+MVDR+tone	70.6

Table 2. *Recognition performance of baseline systems on the Bengali target language*

Results for applying multi-lingual training with shared hidden layers to the DBNF networks are listed in Table 3. Different architectures and input features were applied to the DBNF systems: (1) the baseline DBNFs architecture using MFCC+MVDR+tone as input features (tone no-s); (2) the shifting DBNFs architecture using MFCC+MVDR+tone as input features (tone shift); (3) the baseline DBNFs architecture using MFCC+MVDR as input features (no-t no-s); (4) the shifting DBNFs architecture using MFCC+MVDR as input features (no-t shift);

Additional languages	**tone no-s**	**tone shift**	**no-t no-s**	**no-t shift**
ASM	68.2	67.7	–	–
PUS	68.0	67.7	68.8	68.2
TGL	68.2	67.5	–	–
ASM,PUS	66.5	66.1	67.5	66.9
ASM,PUS,TGL	66.1	65.4	67.1	–
ASM,PUS,TGL,YUE	65.5	65.0	–	–
ASM,PUS,TGL,YUE,VIE	65.1	64.5	–	–

Table 3. *Results for Bengali DBNF systems with shared hidden layers trained on multiple additional languages*

It can be seen that the multi-lingual numbers look most promising. Adding just one more language (ASM, PUS or TGL) the recognition performance was increased by 4.5% relative (a drop in WER from 71.2% to 68.0% or 68.2%). Adding two languages (ASM and PUS) gives a bigger gain (about 1.5% absolute and 2.5% relative) than adding only one language. Adding 3-5 languages also gives a bigger gain but with smaller improvements with each additional language. The proposed architecture with shifting not only gives gains in a mono-lingual but also in multilingual context (from 0.4% to 0.7% absolute improvement). As one can see on the Table 3, adding tonal features gives reductions of about 1% absolute WER compared to the systems using only MFCC+MVDR as input features.

6. CONCLUSION AND FUTURE WORK

With the results obtained above, we have shown that the proposed SDBNF architecture is useful for multilingual DBNF setups trained on not only one language but also on multiple languages for which a larger amount of data might be available. It was shown that the DBNF systems trained with MLBootstrap are better than the ones trained with a flatstart technique. We have also shown that tonal features are helpful for multilingual DBNFs systems. The WER on the the low-resource conversational telephone speech transcription task in Begali was reduced by 17.3% relative when compared to an MFCC baseline system. We achieved this by using additional data from Assamese, Pashto, Tagalog, Turkish and Vietnamese. In the future, we would like to explore how multi-lingual data can be leveraged to improve DBNF network training and to further enhance the architectures suggested.

[3]http://deeplearning.net/software/theano/

7. REFERENCES

[1] T. Schultz and A. Waibel, "Language-independent and language-adaptive acoustic modeling for speech recognition," *Speech Communication*, vol. 35, no. 1-2, pp. 31–51, 2001.

[2] J. Gehring, Y. Miao, F. Metze, and A. Waibel, "Extracting deep bottleneck features using stacked auto-encoders," in *Proceedings of the Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on. IEEE*, Vancouver, CA, 2013, pp. 3377–3381.

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

[4] J. Gehring, Q. B. Nguyen, F. Metze, and A. Waibel, "Dnn acoustic modeling with modular multi-lingual feature extraction networks," in *Proc. Automatic Speech Recognition and Understanding (ASRU)*. Olomouc, Czech Republic: IEEE, 2013, pp. 344–349.

[5] N. T. Vu, W. Breiter, F. Metze, and T. Schultz, "An investigation on initialization schemes for multilayer perceptron training using multilingual data and their effect on asr performance," in *Proceedings of the Interspeech*, 2012.

[6] 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.

[7] K. Vesel, M. Karafit, F. Grzl, 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.

[8] 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 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*. Vancouver, CA: IEEE, 2013.

[9] Z. Tüske, R. Schlüter, and H. Ney, "Multilingual hierarchical mrasta features for asr," in *Proceedings of the Interspeech*, Lyon, France, Aug. 2013, pp. 2222–2226.

[10] 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.

[11] M. N. Murthi and B. D. Rao, "Minimum variance distortionless response (mvdr) modeling of voiced speech," in *Proceedings of the Acoustics, Speech, and Signal Processing, 1997. ICASSP-97., 1997 IEEE International Conference on*, vol. 3. IEEE, 1997, pp. 1687–1690.

[12] F. Metze, Z. Sheik, A. Waibel, J. Gehring, K. Kilgour, Q. Nguyen, and V. Nguyen, "Models of tone for tonal and non-tonal languages," in *Proceedings of the Automatic Speech Recognition and Understanding (ASRU), 2011 IEEE Workshop on*. IEEE, 2013, pp. 261–266.

[13] M. Wölfel and J. McDonough, "Minimum variance distortionless response spectral estimation," *Signal Processing Magazine, IEEE*, vol. 22, no. 5, pp. 117–126, 2005.

[14] 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.

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

[16] F. Grézl, M. Karafiát, S. Kontár, and J. Cernocky, "Probabilistic and bottle-neck features for lvcsr of meetings," in *Proceedings of the Acoustics, Speech and Signal Processing, 2007. ICASSP 2007. IEEE International Conference on*. IEEE, 2007, pp. V–757 – IV–760.

[17] D. Yu and M. L. Seltzer, "Improved bottleneck features using pretrained deep neural networks," in *INTERSPEECH*, 2011, pp. 237–240.

[18] A. Waibel, T. Hanazawa, G. Hinton, K. Shikano, and K. Lang, "Phoneme recognition using time-delay neural networks," *Acoustics, Speech and Signal Processing, IEEE Transactions on*, vol. 37, no. 3, pp. 328–339, Mar 1989.

[19] K. Vesel, M. Karafit, and F. Grzl, "Convolutive bottleneck network features for lvcsr." in *ASRU*, D. Nahamoo and M. Picheny, Eds. IEEE, 2011, pp. 42–47.

[20] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, and P.-A. Manzagol, "Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion," *J. Mach. Learn. Res.*, vol. 11, pp. 3371–3408, Dec. 2010.

[21] 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*, 2001, pp. 214–217.