OPTIMIZING MULTILINGUAL KNOWLEDGE TRANSFER FOR TIME-DELAY NEURAL NETWORKS WITH LOW-RANK FACTORIZATION

Francis Keith, William Hartmann, Man-hung Siu, Jeff Ma, Owen Kimball

Raytheon BBN Technologies, Cambridge MA, USA

{francis.keith, william.hartmann, man-hung.siu, jeff.ma, owen.kimball}@raytheon.com

ABSTRACT

When producing speech-to-text (STT) systems on a lower resource language, it is often beneficial to use knowledge obtained from a significantly larger multilingual dataset. We have seen benefits from using a multilingual TDNN as initialization for training an acoustic model on a target low resource language. In this work, we expand upon recent research that found benefits from applying sequential low-rank factorization (LRF) by extending it to a TDNN acoustic model trained on a large multilingual corpus. We also examine and optimize the knowledge transfer methodology, with the goal of avoiding the loss of useful information from the multilingual initialization during the knowledge transfer process. Our approach limits the updates to the multilingual network parameters during lattice-free maximum mutual information (LF-MMI) training on the target low resource language by fixing the multilingual network parameters and only optimizing the target output layer. The multilingual parameters and new output layer are jointly optimized using the state-level minimum Bayes risk (sMBR) objective function. By combining sequential LRF with this optimization method, we show across low resource target languages an average absolute WER reduction of 1.2%, yielding a better result than our previous best approach.

Index Terms— speech recognition, multilingual training

1. INTRODUCTION

Multilingual knowledge transfer has long been a focus in improving speech-to-text (STT) systems on lower resource languages, especially as the field has transitioned to focus on neural networks [1][2][3][4][5][6][7]. Multilingual training can be thought of as an approach to initialization or pretraining. When the amount of data is limited, this is especially important [8][9]. When the acoustic training data is not limited to a single target language, the amount of the data can be arbitrarily large. Given the large amount of multilingual training data, more sophisticated acoustic models can potentially be used. However, this leads to an optimization problem when all of the model parameters are updated based on the limited amount of data in the target language. One solution is to limit the use of the multilingual training data to a bottleneck feature extractor [10].

An alternative approach is to reduce the number of parameters in the acoustic model. Many of these methods are based on low-rank factorization (LRF), such as singular value decomposition (SVD). Sahraeian and Van Compernolle [11] have recently shown that LRF can be applied in a sequential manner during the training of a multilingual model, reducing the number of parameters that need to be updated during knowledge transfer. They demonstrated that the size of the acoustic model could be significantly reduced while also improving word error rate (WER) on a small dataset of read speech. We extend their technique to a larger conversational telephone speech (CTS) dataset with more than an order of magnitude more multilingual training data. Instead of the traditional feed-forward deep neural network (DNN) acoustic model, we use a time-delay neural network (TDNN) [12] with frame subsampling trained using lattice-free MMI (LF-MMI) [13]. We demonstrate that the technique works on the larger, more difficult dataset using more a more sophisticated acoustic model.

We also perform a more systematic analysis of the optimization process. The typical approach is to update all parameters of the network using a reduced learning rate. We find that only updating the final output layer produces better results. Further gains are obtained by then updating all parameters during sequence training.

In Section 2, we describe our multilingual neural network setup and explain the knowledge transfer process. Section 3 details SVD and a description of the sequential SVD method adapted from the work by Sahraeian and Van Compernolle. In Section 4, we discuss our experiments and results, including the motivation for output layer only LF-MMI optimization, and we present our conclusions in Section 5.

2. MULTILINGUAL TDNN

2.1. Chain TDNN

Recently, BBN has seen a significant amount of success by using TDNNs for STT systems [14]. Additionally, we have seen significant improvements in both speed and WER by

adopting the so-called "chain model" structure [13]. The chain model contains two key features. It uses a two-state topology with a single self-loop that allows traversing a given phone in a single frame. This makes it possible to subsample the input data, allowing this TDNN to be significantly faster at decoding time. The other feature of chain models is the use of lattice-free maximum mutual information (LF-MMI) as a training criterion instead of the typical cross-entropy training. By using chain TDNNs, we are able to get very strong WER results while still running at sub-real time. There have been recent strides in the use of recurrent neural networks, such as long short-term memory neural networks (LSTM) [15], but the run-time decoding of these recurrent networks is generally much slower than that of the chain TDNN. For our purposes, we found the chain TDNN to be an appropriate trade off between decoding speed and word error rate.

2.2. Multilingual training and knowledge transfer

There are numerous ways to leverage data from other languages in an effort to improve the STT system for a target language, with the most prominent being the training of a multilingual bottleneck (BN) feature extractor. In this work we focused on using the multilingual model as an initialization to the acoustic model for the low resource language. This is sometimes referred to as "fine-tuning" or adaptation. The idea is to train a network on a significant amount of data taken from various different languages, and then use that network as an initialization for the target low resource language. In practice the multilingual model should contain as much data as possible, though some work has been done on selection of multilingual data [16]. In our previous work [14], we examined some of the properties of the multilingual corpus, including target language membership in the multilingual training corpus and channel similarities between the data included. It seems that, assuming the multilingual corpus contains sufficient data, these do not have a significant effect on the final results.

The multilingual model is trained with the standard criterion used in training (in our case, LF-MMI) with an output layer that is simply a combination of all phone states for all languages. To tune the multilingual network to the target low resource language, the multilingual output layer is discarded and a new output layer for the target language is created. The language-specific output layer is pretrained with a very small amount of data with the remainder of the network fixed to produce a viable language-specific output layer. Following this, the entire network is trained with the LF-MMI criterion with a smaller learning rate, followed by sMBR training [17] for additional sharpening of the model. In tuning the LF-MMI models, we observed some interesting relationships between the WERs of the LF-MMI trained models and the sMBR trained models, which we will describe further in our experiments.

3. SEQUENTIAL LOW RANK FACTORIZATION

3.1. Singular value decomposition

Historically, singular value decomposition (SVD) has been used in DNNs for parameter reduction. The notion is that SVD can be used to factorize the DNN's large matrix of parameter weights into multiple smaller matrices and thus reduce the overall complexity of the model. In our work, we use SVD as described in [18]. For a given weight matrix A, we can apply SVD as Equation 1, where Σ is a diagonal matrix containing the singular values of A, and U and V contain the left and right singular values of A.

$$A_{m\times n} = U_{m\times n}\Sigma_{n\times n}V^T_{n\times n} \tag{1}$$

Note that this version of SVD in Equation 1 contains no approximation (and thus no parameter reduction). However, as noted in [18], much of the information from the singular values is contained in a subset of the top singular values. Given this, we can use only the top k singular values and replace the equation with the approximation in Equation 2

$$A_{m\times n} \approx U_{m\times k}\Sigma_{k\times k}V^T_{k\times k} \tag{2}$$

Additionally, we can consider $N_{k\times k} = \Sigma_{k\times k}V^T_{k\times k}$ to effectively replace the original weight matrix $A_{m\times n}$ with two smaller weight matrices, $U_{m\times k}$ and $N_{k\times k}$. Assuming that k is sufficiently small, this will yield a parameter reduction and decrease computational complexity.

3.2. Sequential SVD

As observed in [11], using LRF for parameter reduction has some drawbacks. While it reduces model complexity, replacing parameters of a well-trained model with an approximation introduces noise. If enough noise is introduced, the model may not be able to recover with additional training. The goal of performing LRF is two-fold. LRF will clearly yield parameter reduction assuming a sufficiently small k, but by introducing the SVD approximation, we have effectively added noise and perturbed the model. Many methods in DNN training (such as dropout) involve perturbing the model in an effort to regularize and avoid overfitting. LRF should produce a similar effect—by perturbing the model through the LRF approximation and then retraining, we should improve the robustness of the model.

We followed a similar approach as in [11], with the goal of using more data and a more complex model than a feed-forward DNN. We applied SVD to all hidden layers, beginning with the one closest to the output. After applying SVD to each layer, we retrained the model with a subset of the data. Following the factorization and retraining of all hidden layers, the final post-factorization model was trained again.

4. EXPERIMENTS

4.1. Experimental setup

For training the multilingual model, we use the same corpus used in [14]. The corpus contains 1560 hours of CTS data from 11 different languages[1]. The baseline multilingual TDNN was trained on 4 epochs of the data with the LF-MMI criterion, with an initial learning rate of 1×10^{-3} and a final learning rate of 1×10^{-4}. It was trained with the BBN Sage toolkit [19], specifically using the integrated Kaldi [20] portion of the toolkit. The input features used were 40 dimension high resolution MFCC features for the input frame and the frames surrounding it, and 100-dimension i-vectors, for a total input feature vector of size 220. The structure of the model is the standard chain model as described in [13], and the data is subsampled to examine 1 out of every 3 frames. The TDNN structure consisted of 6 hidden layers with 1152 nodes per layer. The splicing configuration for the TDNN is as follows: {0}, {-1,0,1,2},{-3,0,3},{-3,0,3}, {-6,-3,0}, {0}. This can be read as each layer containing the splicing of the layers at a time relative to 0, with 0 being the current frame. So the first hidden layer uses solely the concatenated input feature vector, the second hidden layer uses a concatenation of the first hidden layer outputs at the previous time-step, the current time-step, and the next two time-steps, and so on.

Performing knowledge transfer from the multilingual model to the target language is a straightforward process. The multilingual model is used as an initialization. The output layer is removed and a new output layer is initialized to the target data with a small, fixed number of minibatches. Following this, the model is trained with the LF-MMI criterion for 4 epochs on the target language. The LF-MMI optimization uses a learning rate of 4×10^{-5}. Following LF-MMI training, the model is optimized with 4 epochs of sMBR training using an effective learning rate of 1.25×10^{-6}.

To show the effects on a multitude of languages, we chose 4 CTS target language corpora with varying amounts of data. The target language corpora are as follows: Egyptian (20hrs), Georgian (50hrs), Turkish (83hrs), and Cantonese (110hrs). Note that the Turkish and Cantonese corpora are both included in the multilingual training corpus. However, as observed in [14], membership in the multilingual corpus does not seem to affect knowledge transfer optimization.

4.2. Multilingual sequential SVD model

For the model using sequential SVD, we took an initial version of our multilingual model that had been trained on 2 epochs of data. For SVD we use k=384, $\frac{1}{3}$ of the hidden layer size of the multilingual model. We began by applying SVD between the 5th and 6th hidden layers, and retrained on a portion of the multilingual data. This process was repeated for subsequent hidden layer weight matrices. We used a larger subset of the data than was proposed in [11]. We used 2 epochs of data total for sequential SVD, so after each SVD we retrained the model on $\frac{2}{5}$ of the data. After SVD was applied to all hidden layers the model was trained on 2 further epochs of the data. It is worth noting that our final sequential SVD model is trained on 6 epochs of data—2 initial multilingual LF-MMI epochs, 2 epochs of sequential SVD retraining, and 2 additional epochs of multilingual LF-MMI training following the sequential SVD. Our multilingual baseline was trained with only 4 LF-MMI epochs, but we have observed in practice that additional training to the baseline does not improve the multilingual model with regards to knowledge transfer.

Table 1: *Sequential SVD results compared to the baseline using both LF-MMI and sMBR models (WER)*

Language	Model	LF-MMI	sMBR
Egyptian	Baseline	38.0	36.9
	Seq. SVD	37.7	**36.7**
Georgian	Baseline	42.7	40.1
	Seq. SVD	42.1	**39.6**
Turkish	Baseline	40.5	39.0
	Seq. SVD	40.1	**38.1**
Cantonese	Baseline	40.9	39.8
	Seq. SVD	40.4	**39.0**
Average	Baseline	40.5	39.0
	Seq. SVD	40.1	**38.4**

Table 1 shows the results comparing the sequential SVD models with the baseline models across all languages, along with the average word error rate across all languages. We report word error rate (WER) using both the LF-MMI models and the sMBR models. This shows the importance of sMBR training on top of the standard LF-MMI optimization—even on the baseline, sMBR accounts for a 1.5% absolute reduction to WER on average. This also highlights the improvements of the sequential SVD model, which produces a 0.6% average absolute WER reduction over the baseline.

4.3. Knowledge transfer optimization

As noted in Section 4.2, we used a constant learning rate of 4×10^{-5} for LF-MMI optimization, finding that empirically to yield optimal performance across all languages. In tuning the learning rate, we observed some interesting results that supported a change to our standard optimization. Table 2 shows the comparison between the optimal learning rate and our initial learning rate prior to tuning of 1×10^{-4} on Georgian. Decreasing the learning rate to the optimal value does not necessarily improve the LF-MMI WER significantly—notably, in the sequential SVD model, moving to the optimal learning

[1]The languages and amounts of data for the multilingual corpus are as follows: English (380hrs), Mandarin (250hrs), Spanish (245hrs), Cantonese (110hrs), Pashto (98hrs), Tagalog (90hrs), Vietnamese (90hrs), French (85hrs), Turkish (83hrs), Haitian (80hrs), Swahili (50hrs)

Table 2: *Georgian results from tuning the LF-MMI learning rate (WER)*

Model	LF-MMI learning rate	LF-MMI	sMBR
Baseline	1×10^{-4}	43.4	42.4
	4×10^{-5}	42.7	**40.1**
Seq. SVD	1×10^{-4}	42.3	41.3
	4×10^{-5}	42.1	**39.6**

rate only improves the LF-MMI decoding result by 0.2% absolute. However, the lower learning rate makes the LF-MMI model a better candidate for improvement with sMBR training, as it improves the model by 1.7% absolute over the larger learning rate. The baseline model has a larger gain in LF-MMI WER, but follows a similar trend—the lower learning rate improves the final sMBR WER significantly.

Given the results in Table 2, we tried a different approach to LF-MMI optimization of the target language. It is clear that optimizing the LF-MMI in different ways can significantly improve the final model after sMBR training. We believe that this is in part due to "catastrophic forgetting" [21], where a model trained on one task and adapted to a new task "forgets" how to perform the first task. In this case, optimizing the model to the target language with LF-MMI causes the model to forget much of the useful information learned from multilingual training. Our idea was to use a single epoch of LF-MMI training solely for optimizing the new output layer, leaving the multilingual parameters of the network fixed. For updating the output layer only, we found a much higher learning rate was necessary, settling on 3×10^{-4}. Following this, sMBR is trained with 4 epochs to optimize the parameters of the entire network. sMBR training typically uses a very small effective learning rate (our experiments use 1.25×10^{-6}) and should avoid removing too much multilingual information. sMBR has been shown to be a very effective means of training on top of other objective functions [17][22], and importantly, it is a different optimization criterion from the LF-MMI criterion used to train the multilingual model.

Table 3 shows the tuning on the different types of multilingual knowledge transfer optimization described in 4.3. This includes both sequential SVD models as well as the baseline models, only reporting the final numbers after sMBR training. It is clear that across all languages, there is a gain from doing both output layer only optimization, as well as from sequential SVD. Interestingly, though the average absolute reduction in WER is 1.2%, it appears as though the languages with less data (Egyptian and Georgian) benefit less from doing output layer only lattice-free MMI optimization than the languages with more data. One possibility is that, as noted before, Turkish and Cantonese are present in the multilingual training set. While in previous experiments [14] we observed that this has little effect, it is possible that foregoing all optimization of the network with the lattice-free MMI criterion outside of the output layer could make the matching observations from the multilingual model more significant. These observations could yield potential insight in future work.

Table 3: *Standard LF-MMI optimization compared to output only LF-MMI optimization with sequential SVD after sMBR training (WER)*

Language	Model	Standard optimization (lr: 4×10^{-5})	Output only optimization (lr: 3×10^{-4})
Egyptian	Baseline	36.9	36.8
	Seq. SVD	36.7	**36.3**
Georgian	Baseline	40.1	39.6
	Seq. SVD	39.6	**39.1**
Turkish	Baseline	39.0	38.0
	Seq. SVD	38.1	**37.4**
Cantonese	Baseline	39.8	38.8
	Seq. SVD	39.0	**38.3**
Average	Baseline	39.0	38.3
	Seq. SVD	38.4	**37.8**

Though not our primary objective, using SVD on neural network weight matrices gives us the additional benefit of parameter reduction. The baseline 1152-dimension model contains approximately 40.1 million parameters. Through sequential SVD, we have reduced that number to approximately 28.1 million parameters, a roughly 30% reduction in the total number of parameters.

5. CONCLUSION

In this work we have demonstrated a few different ways to optimize multilingual knowledge transfer. The first was to expand upon the work done in [11], applying it to a larger, more challenging dataset with a more sophisticated model, in this case chain TDNN. We observed an absolute 0.6% average word error rate reduction when using sequential SVD. We also optimized our use of training with the lattice-free MMI criterion, applying a single epoch of target language training and using that solely for training the new output layer. The entire network is then optimized using the sMBR criterion. This yields an additional 0.6% average absolute WER reduction, both on the baseline as well as the sequential SVD model, resulting in an average 1.2% absolute WER reduction when combining the two approaches.

Our previous work [14] focused on combining the effects of knowledge transfer optimization with multilingual BN features. In that work, our best system produced an average result of 38.9% WER on the same set of target languages. We have produced an absolute reduction of WER by 1.1% over those results, and plan to explore combining this work with the BN features explored in that research.

6. REFERENCES

[1] Andreas Stolcke, Frantisek Grezl, Mei-Yuh Hwang, Xin Lei, Nelson Morgan, and Dimitra Vergyri, "Cross-domain and cross-language portability of acoustic features estimated by multilayer perceptrons," in *Proceedings of ICASSP*, 2006.

[2] Samuel Thomas, Sriram Ganapathy, and Hynek Hermansky, "Multilingual MLP features for low-resource LVCSR systems," in *Proceedings of ICASSP*, 2012.

[3] Kate M Knill, Mark JF Gales, Shakti P Rath, Philip C Woodland, Chao Zhang, and S-X Zhang, "Investigation of multilingual deep neural networks for spoken term detection," in *Proceedings of the IEEE Workshop on Automatic Speech Recognition and Understanding*, 2013.

[4] Jui-Ting Huang, Jinyu Li, Dong Yu, Li Deng, and Yifan Gong, "Cross-language knowledge transfer using multilingual deep neural network with shared hidden layers," in *Proceedings of ICASSP*, 2013.

[5] Quoc Bao Nguyen, Jonas Gehring, Markus Muller, Sebastian Stuker, and Alex Waibel, "Multilingual shifting deep bottleneck features for low-resource ASR," in *Proceedings of ICASSP*, 2014.

[6] Tom Sercu, George Saon, Jia Cui, Xiaodong Cui, Bhuvana Ramabhadran, Brian Kingsbury, and Abhinav Sethy, "Network architectures for multilingual speech representation learning," in *Proceedings of ICASSP*, 2017.

[7] Martin Karafiát, Murali Karthick Baskar, Pavel Matejka, Karel Veselỳ, and František Grézl, "2016 BUT Babel system: Multilingual BLSTM acoustic model with i-vector based adaptation," in *Proceedings of INTERSPEECH*, 2017.

[8] Ngoc Thang Vu, Wojtek Breiter, Florian Metze, and Tanja Schultz, "An investigation on initialization schemes for multilayer perceptron training using multilingual data and their effect on ASR performance," 2012.

[9] Xavier Glorot and Yoshua Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, 2010.

[10] K. Vesely, M. Karafiát, F. Grézl, M. Janda, and E. Egorova, "The language-independent bottleneck features," in *Proceedings of SLT*, 2012.

[11] Reza Sahraeian and Dirk Van Compernolle, "Exploiting sequential low-rank factorization for multilingual DNNs," in *Proceedings of ICASSP*, 2017.

[12] Alex Waibel, Toshiyuki Hanazawa, Geoffrey Hinton, Kiyohiro Shikano, and Kevin J Lang, "Phoneme recognition using time-delay neural networks," *IEEE transactions on acoustics, speech, and signal processing*, vol. 37, no. 3, pp. 328–339, 1989.

[13] Daniel Povey, Vijayaditya Peddinti, Daniel Galvez, Pegah Ghahremani, Vimal Manohar, Xingyu Na, Yiming Wang, and Sanjeev Khudanpur, "Purely sequence-trained neural networks for ASR based on lattice-free MMI," in *Proceedings of INTERSPEECH*, 2016.

[14] Jeff Ma, Francis Keith, Tim Ng, Man-Hung Siu, and Owen Kimball, "Improving deliverable speech-to-text systems with multilingual knowledge transfer," in *Proceedings of INTERSPEECH*, 2017.

[15] Hasim Sak, Andrew Senior, and Franoise Beaufays, "Long short-term memory based recurrent neural network architectures for large vocabulary speech recognition," in *Proceedings of INTERSPEECH*, 2014.

[16] Ekapol Chuangsuwanich, Yu Zhang, and James Glass, "Multilingual data selection for training stacked bottleneck features," in *Proceedings of ICASSP*, 2016.

[17] Karel Veselỳ, Arnab Ghoshal, Lukás Burget, and Daniel Povey, "Sequence-discriminative training of deep neural networks," in *Proceedings of INTERSPEECH*, 2013.

[18] Jian Xue, Jinyu Li, and Yifan Gong, "Restructuring of deep neural network acoustic models with singular value decomposition," in *Proceedings of INTERSPEECH*, 2013.

[19] Roger Hsiao, Ralf Meermeier, Tim Ng, Zhongqiang Huang, Maxwell Jordan, Enoch Kan, Tanel Alume, Jan Silovsky, William Hartmann, Francis Keith, Omer Lang, Manhung Siu, and Owen Kimball, "Sage: The new BBN speech processing platform," in *Proceedings of INTERSPEECH*, 2016.

[20] Daniel Povey, Arnab Ghoshal, Gilles Boulianne, Lukas Burget, Ondrej Glembek, Nagendra Goel, Mirko Hannemann, Petr Motlicek, Yanmin Qian, Petr Schwarz, Jan Silovsky, Georg Stemmer, and Karel Vesely, "The Kaldi speech recognition toolkit," in *Proceedings of ASRU*, 2011.

[21] Ian J. Goodfellow, Mehdi Mirza, Da Xiao, Aaron Courville, and Yoshua Bengio, "An empirical investigation of catastrophic forgetting in gradient-based neural networks," in *arXiv preprint arXiv:1312.6211*, 2013.

[22] Jeremy HM Wong and Mark JF Gales, "Sequence student-teacher training of deep neural networks," in *Proceedings of INTERSPEECH*, 2016.