Multilingual Multi-Task Learning for Low-Resource Acoustic Modeling

Josh Meyer[1]

[1]University of Arizona

joshua.richard.meyer@gmail.com

Abstract

Index Terms: speech recognition, multi-task learning, acoustic modeling

1. Introduction

Previous work has shown that performance for a low-resource language on speech recognition can be improved by adding training data from another, resource-rich language. Typically, data from another language is added as a separate task in the Multi-Task Learning framework via an additional output layer (Caruana 1997).

The targets for this addititional language have always been states of context-dependent triphones, defined by some tree clustering algorithm. This current research builds off the intuition that triphones encode information which is too fine-grained to be maximally useful for language-transfer. Using a higher-level of linguistic abstraction (eg. the monophone), we are able to better extract the kind of language-general information useful in training an acoustic model for some target language.

To put it another way, if we want to lower Word Error Rates for a language like Urdu, learning to distinguish different versions of English [th] in context is probably not very useful. A better way to make use of English data would be to focus on distinguishing more common linguistic contrasts like [p vs b]. In adding an additional language as an auxialilary task, it would be better to focus on distinctions which are robust and will transfer well to a new, target language.

The tasks are created by redefining the parameters of the HMM-GMM system used to bootstrap a DNN-hybrid system, such that the phonetic decision tree is cut short.

The target language is Kyrgyz, and the source language is English. Both data come from audiobooks, English being from LibriSpeech and Kyrgyz from the Bizdin.kg project.

2. Background Literature

Past attempts at MTL

3. DNN-Hybrid Training as Model Transfer

The standard DNN-Hybrid approach requires the GMM-HMM system to provide the labels for supervised training. This reliance of the DNN on GMM alignments is actually a form of model transfer, where the DNN is trained to perform the extact same classification as its GMM predecessor. The DNN not only learns the frame alignments from the individual GMMs, but also the DNN indirectly learns the structure of the phonetic decision tree used to define the tied-state system. This is because the output layer of the DNN is trained to predict targets which were defined via leaves of the decision tree.

Given that standard triphones encode very fine-grained information which may not help performance on a target language, the following experiments investigate GMM→DNN model transfer at a higher level for an additional source language.

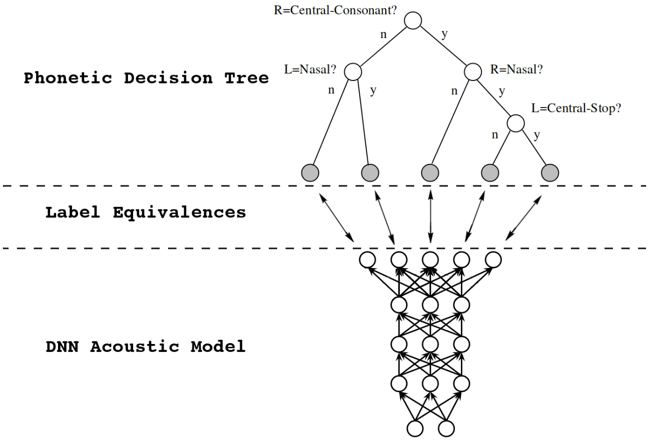

Figure 1: *GMM→DNN Model Transfer*

4. Experiments

This work investigates the application of MTL technique to low resource acoustic modeling. All experiments relate to simulate a common development scenario: there exists little transcribed data for the target language, but lots of data in some source language.

The following experiments tease out (1) the level of detail at which the source language should be modeled and (2) the amount of weighting which should be given to the target language training examples.

The first point of interest: level of detail at which the source language should be modeled. This is investigated via addition of multiple tasks to the TDNN during backprop. Intuitively, we can model the source language with lots of contextual detail (ie. the triphone), with abstracted, context-independent detail (ie. the monophone), or somewhere in between (ie. the half-phone). Given the traditional ASR training pipeline, investigating these levels of representation are easily acheived via the phonetic decision tree.

The second point of interest: relative weighting of target:source training examples. It is clear that if we train multiple languages in parallel, the source language (given many more training samples) will dominate the target language in the fight for influence during backprop. This could be a good thing or a bad thing, and to investigate it further, I examine the following weighting schemes: 1:2 target:source; 1:1 target:source; 2:1 target:source. These weights are instantiated during backprop as a weight to the gradient for each training example. For example, given 1000 hours of source language and 1 hour of target language, to acheive a 1:1 ratio in backprop, I would multiple the gradient from the target language by 1000.

4.1. Data

Two speech corpora are used in the following experiments:

1. ≈ 5 hours of LibriSpeech (4.86 hours)
2. ≈ 1.5 hours of Kyrgyz audiobook (1.59 hours)

4.2. Model Building

4.2.1. Decision Trees

All models were build using the Kaldi `nnet3` approach. The scripts used in this paper can be found at (XXX). These scripts are based on the offical repo multilingual Babel scripts here (XXX).

The initial decision tree parameters were found independently for English and Kyrgyz, and were chosen to maximize performance on a held out test set of data for each language.

In GMM training, monophones (for each language) were allotted 1,000 Gaussian components, and trained over 25 iterations of EM. These monophones were then expanded into context-dependent triphones via a phonetic decision tree, with a maximum of 2,000 leaves & 5,000 Gaussians (LibriSpeech reached 1584 leaves, and Kyrgyz reached 752). The resulting tied-state clusters (ie. leaves) are then trained as context-dedendent triphones over 25 iterations of EM.

4.2.2. Multi-Task Neural Net Acoustic Models

Given the alignments from the GMM-HMM models, a 5-layer, 500-dimensional TDNN is trained over 5 epochs of backprop on a single GPU instance.

Each auxiliary task is implemented as a separate output layer along with a separate, penultimate hidden layer. All other hidden layers of the TDNN are trained in parallel.

Baseline

All the following architectures will be compared to the performance of the following baseline.

To account for any advantage mutliple output layers may bring about, the baseline also contains two output layers, where the tasks are identical. In this way, random initializations in the weights and biases for each task are accounted for.

During testing, *only one* of the tasks (ie. the main task) is used. The additional tasks are dropped and the baseline Kyrgyz triphones are used in decoding. This highlights the purpose of the extra tasks: to force the learning of robust representations in the hidden layers during training. The tasks may in fact not be the best option for final classification; they serve as "training wheels" which are then removed once the net is ready.

Auxiliary Tasks

The auxiliary tasks all related to the English language data from the LibriSpeech corpus. Investigating the intuition that labels generated by a standard triphone phonetic decision tree are not the best representation of data for transfer learning, the auxiliary tasks here investigate different levels in the decision tree's branches. By forcing the neural net to recognize higher levels in the tree, we will learn representations which are more abstract, and therefore more likely to be relevant multi-lingually.

Each of the following tasks were trained on the 5 hour section of LibriSpeech corpus. They are included as an extra output layer in the TDNN.

Table 1: *Auxiliary Tasks*

Logical Tree Part	**Level of Phonetic Detail**	**№ of Tasks**
Roots	Monophones	1
Branches	Half-phones	1
Leaves	Triphones	1
Lower Tree	Monophones + Half-phones	2
Upper Tree	Half-phones + Triphones	2
Whole Tree	Monophones + Half-phones + Triphones	3

The addition of each above task adds approximately 5 hours of training data to the standard training of a Single Task Model on Kyrgyz. As such, a weighting procedure was used to balance the relative influence of source vs. target training data on backprop. For example, to reach a one-to-one ratio, where one hour of Kyrgyz is equal to one hour of English, I multiplied every Kyrgyz gradient by 3.06.

Table 2: *Target:Source Data Weighting Scheme*

Target:Source Ratio	**Target Weight**
1:2	1.53x
1:1	3.06x
2:1	6.11x

4.3. Results

All results are performed on the same held-out section of Kyrgyz audiobook. The bigram language model, lexicon, and main-task decision tree are build into a standard decoding graph in the traditional Kaldi style. Decoding is performed with a bigram backoff language model trained on a Wikipedia Kyrgyz dump, and contains, 103,998 unigrams and 56,6871 bigrams.

5-layer, 500 dimensial hidden layer, 10 epoch results:

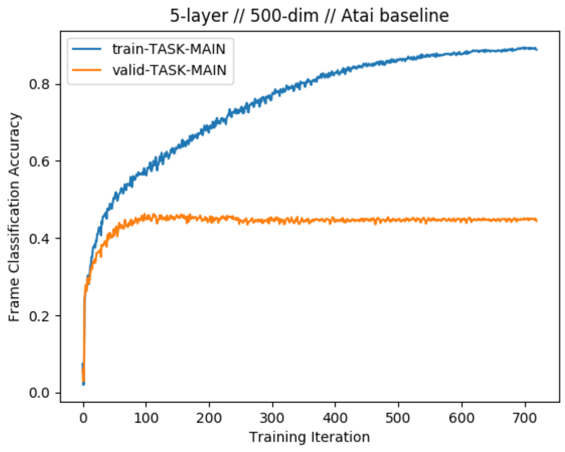

Figure 2: *Baseline Model*

As we can see, the baseline model overfits to the training data after about 250 iterations. The performance for the final model (clearly would be beat out if we used early stopping) is 53.66% WER on decoding the held-out test data.

Above, we see the addition of triphones as an auxiliary task from the LibriSpeech corpus. Overfit to Atai after 500 iterations. The performance for the final model (clearly would be

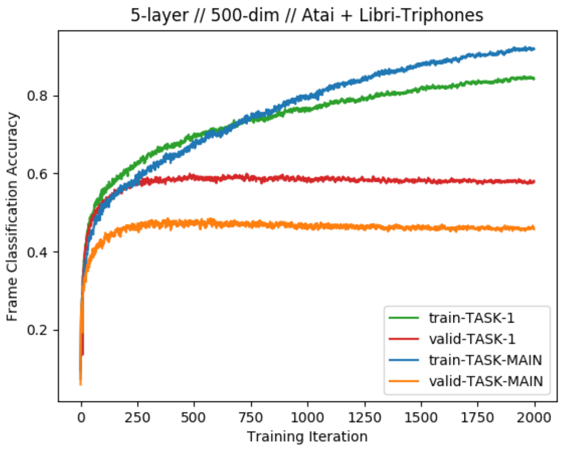

Figure 3: *Aux Task == LibriSpeech Triphones*

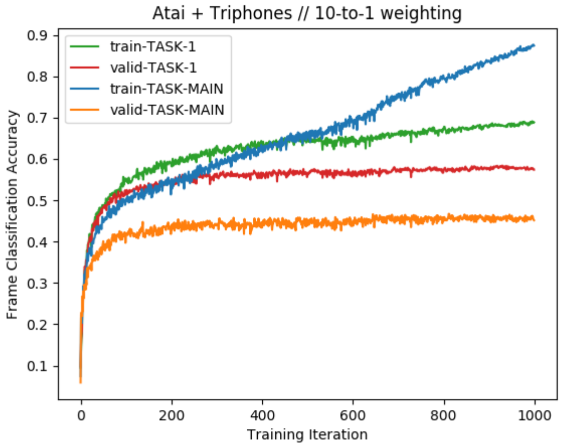

Figure 4: *Aux Task == LibriSpeech + Triphones // 10-to-1*

beat out if we used early stopping) is 49.85% WER on decoding the held-out test data.

Finally, monophones from the LibriSpeech corpus. Overfit to Atai after 500 iterations. The performance for the final model (clearly would be beat out if we used early stopping) is 51.32% WER on decoding the held-out test data.

Here we are with the last model:

Table 3: *5-to-1 Weighting // 5-layer // 10 epoch // 500-dim*

Tasks	**WER%**
Atai + Atai (STL Baseline)	51.54%
Atai + Libri-Triphones	52.29 %
Atai + Libri-Monophones	50.93 %
Atai + Libri-Monophones + Libri-Triphones	50.44 %

Takeaways from 500-dim // 5-epoch // 2-to-1 weighting:

1. Addition of LibriSpeech beats out Kyrgyz-only.
2. Triphones work better than monophones
3. Both languages / tasks overfit

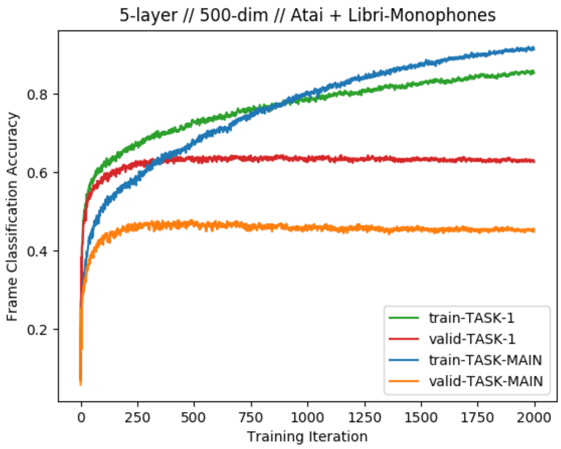

Figure 5: *Aux Task == LibriSpeech Monophones*

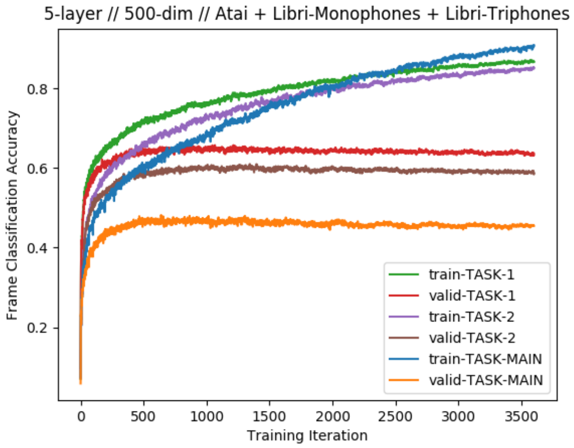

Figure 6: *Aux Task == LibriSpeech Monophones + Triphones*

Table 4: *10-to-1 Weighting // 5-layer // 10 epoch // 500-dim*

Tasks	**WER%**
Atai + Atai (STL Baseline)	51.54%
Atai + Libri-Triphones	51.51 %
Atai + Libri-Monophones	53.66 %
Atai + Libri-Monophones + Libri-Triphones	53.46 %

4. best triphone model beats best baseline (ie. early stopping)
5. atai overfit slower with additional tasks
6. atai overfits with monophones earlier, I think because it's an easier task
7. 10-to-1 weighting on 5 epochs seems like we've overfit the data, but validation is STILL increaing... try to run on 10 epochs and see what happens. 500 dim, 5 layer, aux task == triphones.
8. try 5-to-1, because 2-to-1 seems too weak, and 10-to-1 seems too strong

9. only train on 30 minutes of Krygyz... more likely to find an effect

5. Discussion

6. Conclusions

7. Acknowledgements