TRAINING TIME REDUCTION AND PERFORMANCE IMPROVEMENTS FROM MULTILINGUAL TECHNIQUES ON THE BABEL ASR TASK

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

Institute for Anthropomatics, Karlsruhe Institute of Technology, Germany

ABSTRACT

In the IARPA sponsored program BABEL we are faced with the challenge of training automatic speech recognition systems in sparse data conditions in very little time. In this paper we show that by using multilingual bootstrapping techniques in combination with multilingual deep belief bottle neck features that are only fine tuned on the target language the training time of an LVCSR system can be essentially halved while the word error rate stays the same. We show this for recognition systems on Tagalog, making use of multilingual systems trained on the other four languages of the Babel base period: Cantonese, Pashto, Turkish, and Vietnamese.

Index Terms— automatic speech recognition, multilingual speech recognition, rapid system development

1. INTRODUCTION

In the IARPA sponsored project BABEL[1] we are faced with the challenge of rapidly creating keyword search systems in new languages with only very little training data in the new language. During the end of the project the amount of available acoustic model training data will be less than 10 hours and the allowed training time will be only one week.

Since state-of-the-art keyword spotting systems make use of the output of *large vocabulary continuous speech recognition* (LVCSR) systems, this means that we need to be able to train *automatic speech recognition* (ASR) systems with little training data in a rapid manner. In this paper we will focus on the training of the *acoustic model* (AM) and *pre-processing* of the speech recognition system.

Past research has shown that the use of multilingual acoustic modeling techniques can be used to initialize acoustic models based on *Hidden Markov Models* (HMMs) using *Gaussian Mixture Models* (GMMs) to estimate state emission probabilities [1].

Recently, the concept of multilingual speech recognition has also been extended to *deep belief neural networks* (DNNs) [2] that are being used in speech recognition systems, either for feature extraction in the form of *deep belief bottle neck features* (DBNFs) [3] or for estimating HMM state emission probabilities in hybrid HMM/DNN systems [4].

In this paper we present a detailed timing study that compares the training time for speech recognition systems on the BABEL task for the language Tagalog, comparing the use of multilingual models for model initialization vs. training solely on the available Tagalog training data. We also compare the use of DBNFs trained solely on Tagalog vs. multilingual DBNFs trained on four other languages.

We will show that with the help of multilingual modeling techniques, both for model initialization and DBNF training, the training time can be approximately halved while reaching either comparable or even better *word error rates* (WERs), depending on the amount of available training data.

2. FLAT START MODEL INITIALIZATION VS. MULTILINGUAL BOOTSTRAPPING

2.1. Flat Start

Our baseline model initialization technique works solely on the training data available in the target language. In order to train an HMM/GMM acoustic model with the help of the *maximum likelihood* (ML) training criterion using the *expectation maximization* (EM) algorithm, one needs an initial set of parameters for the HMM and GMMs of the model.

To do this task we have implemented a flat start training procedure that estimates good initial parameters from the training data, instead of relying on randomly initialized parameters.

As pre-processing for our flat-start training we use a standard MFCC front-end, with frame stacking and *linear discriminant analysis* (LDA) for dimensionality reduction. First, we divide the training data into speech and non-speech portions using a simple energy based classifier. We then estimate the mean of all speech and all non-speech frames in the

Supported 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{101-v0.4c,102b-v0.4,103b-v0.4,104b-v0.4bY,105b-v0.4,106-v0.2f,107b-v0.7}.

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

978-1-4799-2893-4/14/$31.00 ©2014 IEEE

data. For every model that belongs to a real speech phoneme, we create one single Gaussian with the mean value set to the mean of the speech frames and one Gaussian for non-speech models, such as silence and noise, with its mean set to the mean of the non-speech MFCCs.

With this initial set of models we create a forced alignment of the training data. From this forced alignment data we now successively create more complex, context-independent models by iterating the following training procedure: a) estimate an LDA transform, b) extract data samples, c) incremental growing of Gaussians training, d) write new forced alignments with the newly estimated models.

In the first iteration of this procedure we set the maximum number of Gaussians in the incremental growing of Gaussians training to one and double it in every iteration. In total we perform 8 iterations, so that the final model after this flat start procedure has 128 Gaussian components per model.

2.2. Bootstrapping

For bootstrapping acoustic models in a new language we use the technique ML-Mix [1]. ML-Mix relies on the availability of a language independent phoneme annotation scheme, such as SAMPA, X-SAMPA, IPA or GlobalPhone. When training an acoustic model it pools the training data from all training languages to train models based on the common phoneme set representation. Phonemes in different languages that share the same annotation also share all the corresponding training data from those languages.

With a context-independent acoustic model trained via ML-Mix it is straight-forward to initialize a context-independent acoustic model in a new language. Parameters from the multilingual model simply need to be copied over to the models of the new language that share the same phoneme notation. In case that a phoneme in the target language exists that is not represented by the multilingual model, one needs to back off to the model that is closest to the missing phoneme. This back-off can be done manually, by using a set of rules that operate on the common phoneme set, or in a data-driven manner [5].

3. MONOLINGUAL VS. MULTILINGUAL DEEP BOTTLE NECK FEATURES

The use of *deep bottle neck features* (DBNFs) for automatic speech recognition has been intensively studied over the last couple of years [3, 6, 7]. For DBNFs a deep-belief network with several hidden layers and one bottleneck layers is trained, that classifies extracted feature vectors as, e.g., phonemes, context-dependent phonemes, or even model states. The layers of the DBNF are usually pre-trained, either by using *Restricted Boltzmann Machines* (RBMs) or denoising auto-encoders. After that back-propagation training, in one of several possible variants, is applied which is sometimes called fine-tuning. For our experiments we use our set-up from [8].

Deep neural networks are well suited for intermediate representations in their hidden layers of different tasks. Multilingual speech recognition is one instance of this where different languages share different sounds and might differ in others. Training multilingual DBNFs can either be done by using one shared phoneme set [9], as it is done for ML-Mix, or by using different language dependent output layers, one for every language [10, 11, 12, 13].

4. EXPERIMENTAL SET-UP

4.1. Data

We performed our experiments with the help of the data corpora provided during the base period of the Babel project. The corpora cover the languages Cantonese, Tagalog, Turkish, Pastho, and Vietnamese. For all languages, two conditions with respect to the amount of training data exist, the *full language pack* (full LP) and the *limited language pack* (limited LP). For the full language pack of every language approx. 70–80 hours of data are available, while for the limited language pack approx. 10 hours of training data are available. For our experiments we used the full language packs of Cantonese, Turkish, Pashto and Vietnamese for training our multilingual model for bootstrapping. For Tagalog we used both language packs, full and limited, to measure the influence of the amount of training data in the target language, when creating a new ASR system for it.

The data consists of telephone conversations recorded over land line and mobile telephones directly in the countries in which the languages are spoken. The recordings were made in varying surroundings such as offices, in the street or in the car.

Table 2 shows the exact amounts of training data used in the experiments in this paper. For testing we used the development set of Tagalog which is 10 hours in length.

Language	Full Pack	Limited Pack
Cantonese	68	—
Pashto	79	—
Turkish	72	—
Vietnamese	79	—
Tagalog	73	9

Table 2. Size of the training data for every language in hours

4.2. System Training

The acoustic models of our speech recognition systems are all based on HMMs with a left-to-right topology without skip-states. All phonemes are modeled with three states and gen-

Limited LP						Full LP					
Flat Start			Multilingual Bootstrap			Flat Start			Multilingual Bootstrap		
Step	Time	WER	Step	Training	WER	Step	Training	WER	Step	Time	WER
Flat Start	0.8	88.9	Bootstrap	0.0	91.9	Flat Start	2.3	87.8	Bootstrap	0.0	91.6
CI ML	2.4	86.3	CI ML	2.0	85.4	CI ML	13.6	83.6	CI ML	9.7	81.7
CD ML	2.8	79.6	CD ML	2.4	78.0	CD ML	11.8	67.7			
DBNF Mono	10.7	70.0	DBNF Mono	10.3	69.6	DBNF Mono	36.6	56.7	DBNF Mono	33.5	55.4
			DBNF Multi	3.7	69.9				DBNF Multi	19.8	55.9

Table 1. Training times in hours and word error rates for the Tagalog limited and full LP training condition

eralized quinphones were clustered with the help of a regression tree. For training our systems that do not use DBNFs we use a standard MFCC front-end that extracts 13 dimensional features every 10ms and uses a window length of 16ms. 15 consecutive frames are stacked and the dimensionality of the resulting vector is reduced to 42 using LDA. The pronunciation dictionaries of our systems were all provided by IARPA in the language packs, and only the pronunciations given in these dictionaries were used in training and testing.

The acoustic model training was done in several stages. First, initial *context-independent* (CI) models were created either with the help of the flat start procedure described in Section 2.1 or by multilingually bootstrapping as described below.

Then six iterations of the following training scheme were performed: a) writing forced alignments with the current model set, b) estimating the LDA transformation, c) extracting training samples for every model, d) incremental growing of Gaussians training, e) semi-tied covariance training[14].

After these six iterations generalized quinphone models were created with the help of a cluster tree. For the full LP condition 10,000 models were clustered, for the limited LP 2,500.

For training the context dependent models we used the same procedure as for training one iteration of context-independent models.

For decoding a tri-gram language model using modified Kneser-Ney back-off was estimated on the transcriptions of the training data, with all words of the training transcriptions in the vocabulary.

4.3. Multilingual Bootstrap

For bootstrapping we trained a context-independent multilingual acoustic model using the technique ML-Mix as described in Section 2.2. The model was trained on the languages Cantonese, Pashto, Turkish, and Vietnamese. Since all the dictionaries that were provided with the language packs by IARPA were given in X-SAMPA notation, we used that as common phoneme set. The multilingual model was trained with forced alignments that came from monolingual recognizers trained for the individual languages.

We then bootstrapped a context-independent acoustic model for Tagalog by copying over the corresponding parameters from the multilingual modeling. Out of the 45 Tagalog phonemes in our system, seven were not covered by the multilingual model. For these we fell back to the closest model from the multilingual model by performing a manual mapping based on the articulatory features associated with the phonemes in the respective models.

4.4. Bottle Neck Features

For training our monolingual bottle neck features on Tagalog we used the set-up from [8]. For training the multilingual DBNFs we followed [15], however without the shifting part. That is, we pre-train the hidden layers as auto-encoders on all languages, except Tagalog, and do the same for the fine-tuning, but with language specific output layers. As input features to the network we used stacked MFCC, *Minimum Variance Distortionless Response* (MVDR), pitch, and *fundamental frequency variation* (FFV) features as described in [16].

5. EXPERIMENTAL RESULTS

We performed our experiments on the 10h Tagalog development set provided by IARPA. The training times and WERs of all systems that we trained and tested are summarized in Table 1 for the limited LP and full LP conditions.

The timings of the trainings were measured on a server with two AMD Opteron 6134 processors with 8 cores each and a frequency of 2.3GHz, and 126GB of main memory. All training data was stored on a local hard disk.

5.1. Baseline System

As a baseline we trained a Tagalog system both for the full and for the limited training data condition using our flat start training procedure described in Section 2.1, followed by the regular training procedure described in Section 4.2 which resulted in two maximum likelihood trained context-dependent systems. On top of that we then trained a DBNF system only on the Tagalog data as described in Section 4.4. For the full language pack condition the context-dependent Tagalog system took 15.6h to train and resulted in a WER of 69.9%. For the limited condition the training took 2.8h and yielded a WER of 79.6%. The extension to a DBNF system raised the total training time to 36.6h for the full LP condition and lowered the WER to 56.7%, while for the limited LP system the total training time was 10.7h and the WER achieved was 70.0%.

5.2. Multilingually Bootstrapped Systems

In comparison we used the multilingual bootstrap method described in Section 4.3 to initialize the models. After that we again performed the regular training procedure to obtain two context-dependent maximum likelihood trained models, one for the full LP and one for the limited LP. The training time for the limited LP models was 2.4 hours and led to a WER of 78.0%. For the full LP the training time was 11.8 hours and resulted in a WER of 67.7%. So, while the training time was only reduced by 14% relative, 24% respectively, when initializing the models with the multilingual bootstrap, the word error rate is in both cases significantly lowered. When considering the absolute training times, especially for the limited LP condition, the gain in training time from the multilingual bootstrap is thus not that important in practice, but the reduction in word error rates are of more interest.

5.3. Monolingual vs. Multilingual DBNF Training

In a next step, we now trained DBNF networks and systems on top of them. For the models trained from the flat start initialization we only trained one set of DBNFs that were trained only on the Tagalog data, by first pre-training its layers and then fine-tuning it on Tagalog. For the models initialized with multilingual bootstrapping we trained the same monolingual DBNFs and an additional set of DBNF networks, by taking the multilingually trained DBNF described in Section 4.4 and only fine-tuning it on the Tagalog data. Since most of the time, when training the DBNF networks, is spent on the pre-training, this method speeds up the training time significantly.

For the models initialized with flat-start the total training time of the system with the monolingual DBNF is 10.7h for the limited LP with a word error rate of 70.0%. For the full LP the training time is 36.6h with a word error rate of 56.7%. When training a monolingual DBNF on top of the multilingually bootstrapped models, the training time for the limited LP systems raises to 10.3h and a WER of 69.6% is reached.

However, when using the multilingually trained DBNFs that are only fine-tuned on Tagalog, and applying them on top of the multilingually bootstrapped models, the training time for limited LP is reduced to 3.7h while the WER stays at 69.9%, while for the full LP the training time is reduced to 19.8h with a WER of 55.9%.

So, when using multilingually boostrapped models in combination with multilingual DBNFs, the training time for the limited LP is reduced by 65% compared to the monolingually trained models, while the WER essential stays the same (-0.1% abs.). For the full LP the training time is reduced by 46% while the WER is even decreased by 0.8% abs.

Figures 1 and 2 visualize these reductions in training time with essentially unchanging word error rates.

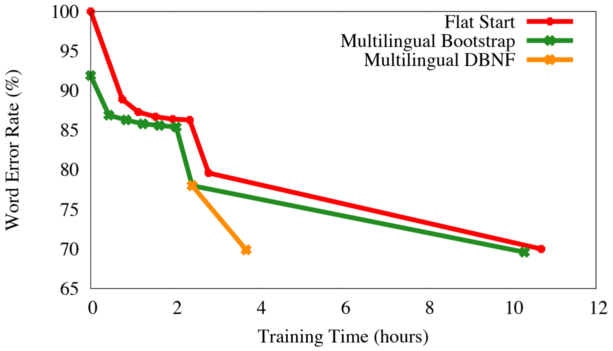

Fig. 1. Training Time plotted against WER for Flat Start training vs. Multilingual Bootstrapping for limited LP

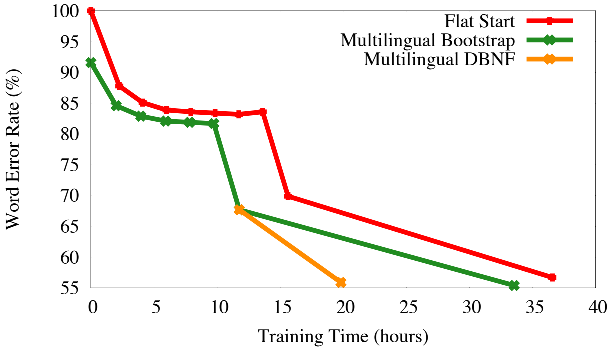

Fig. 2. Training Time plotted against WER for Flat Start training vs. Multilingual Bootstrapping for full LP

6. CONCLUSION

In this paper we have measured the influence of using multilingually bootstrapped acoustic models vs. models obtained by flat start training on the training time of a LVCSR system for Tagalog on the Babel task, as well as on the resulting word error rate of the recognition system. We further measured the differences in training time and WER when using monolingual DBNFs vs. multilingual ones that were only fine tuned on Tagalog. By combining multilingually bootstrapped models with multilingual DBNFs we were able to reduce the training time by 46-65% while the word error rate remained essentially the same or was even slightly lowered, depending on the data condition.

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, August 2001.

[2] Geoffrey E. Hinton, Simon Osindero, and Yee-Whye Teh, "A Fast Learning Algorithm for Deep Belief Nets," *Neural Computation*, vol. 18, no. 7, pp. 1527–1554, July 2006.

[3] Dong Yu and Michael L. Seltzer, "Improved Bottleneck Features Using Pretrained Deep Neural Networks," in *Proceedings of INTERSPEECH*, Florence, Italy, August 2011.

[4] G.E. Dahl, Dong Yu, Li Deng, and A. Acero, "Context-Dependent Pre-Trained Deep Neural Networks for Large-Vocabulary Speech Recognition," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 20, no. 1, pp. 30–42, January 2012.

[5] Sebastian Stüker, "Integrating Thai Grapheme based Acoustic Models into the ML-Mix framework - for language independent and cross-language ASR," in *Proceedings of the First International Workshop on Spoken Languages Technologies for Under-resourced languages (SLTU)*, Hanoi, Vietnam, May 2008.

[6] L. Mangu, Hong-Kwang Kuo, S. Chu, B. Kingsbury, G. Saon, Hagen Soltau, and F. Biadsy, "The IBM 2011 GALE Arabic Speech Transcription System," in *Proceedings of the ASRU*, Waikoloa, HI, USA, December 2011.

[7] T.N. Sainath, B. Kingsbury, and B. Ramabhadran, "Auto-Encoder Bottleneck Features Using Deep Belief Networks," in *Proceedings of the ICASSP*, Kyoto, Japan, March 2012.

[8] Jonas Gehring, Yajie Miao, Florian Metze, and Alex Waibel, "Extracting Deep Bottleneck Features Using Stacked Auto-Encoders," in *Proceedings of the ICASSP*, Vancouver, Canada, May 2013.

[9] Ngoc Thang Vu, Wojtek Breiter, Florian Metze, and Tanja Schultz, "Initialization Schemes for Multilayer Perceptron Training and their Impact on ASR Performance using Multilingual Data," in *Proceedings of the INTERSPEECH*, Portland, Oregon, September 2012.

[10] K. Veseley, M. Karafiat, F. Grezl, M. Janda, and E. Egorova, "The Language-Independent Bottleneck Features," in *Proceedings of the SLT*, Miami, FL, USA, December 2012.

[11] 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.

[12] Stefano Scanzio, Pietro Laface, Luciano Fissore, Roberto Gemello, and Franco Mana, "On The Use of a Multilingual Neural Network Front-End," in *Proceedings of the INTERSPEECH*, Brisbane, Australia, September 2008.

[13] Arnab Ghoshal, Pawel Swietojanski, and Steve Renals, "Multilingual Training of Deep-Neural Networks," in *Proceedings of the ICASSP*, Vancouver, Canada, 2013.

[14] M.J.F. Gales, "Semi-Tied Covariance Matrices for Hidden Markov Models," Tech. Rep., Cambridge University, Engineering Department, February 1998.

[15] Jonas Gehring, Quoc Bao Nguyen, Florian Metze, and Alex Waibel, "DNN Acoustic Modeling with Modular Multi-Lingual Feature Extraction Networks," in *Proceedings of the ASRU*, Olomuc, Czech Republic, December 2013.

[16] Quoc Bao Nguyen, Jonas Gehring, Kevin Kilgour, and Alex Waibel, "Optimizing Deep Bottleneck Feature Extraction," in *Proceedings of the RIVF*, Hanoi, Vietnam, 2013.