ANALYSIS OF MULTILINGUAL BLSTM ACOUSTIC MODEL ON LOW AND HIGH RESOURCE LANGUAGES

Martin Karafiát, Murali Karthick Baskar, Karel Veselý, František Grézl, Lukáš Burget, Jan Černocký

Brno University of Technology Speech@FIT and IT4I Center of Excellence
Brno, Czech Republic

ABSTRACT
The paper provides an analysis of automatic speech recognition systems (ASR) based on multilingual BLSTM, where we used multi-task training with separate classification layer for each language. The focus is on low resource languages, where only a limited amount of transcribed speech is available. In such scenario, we found it essential to train the ASR systems in a multilingual fashion and we report superior results obtained with pre-trained multilingual BLSTM on this task. The high resource languages are also taken into account and we show the importance of language richness for multilingual training. Next, we present the performance of this technique as a function of amount of target language data. The importance of including context information into BLSTM multilingual systems is also stressed, and we report increased resilience of large NNs to overtraining in case of multi-task training.

Index Terms— Automatic speech recognition, Multilingual neural networks, Bidirectional Long Short Term Memory

1. INTRODUCTION AND PRIOR WORK
Quick delivery of an automatic speech recognition (ASR) system for a new language is one of the challenges in the community. Such scenarios call not only for automated construction of systems, that have been carefully designed and crafted “by hand”, but also for effective use of available resources. Without any question, the data collection and annotation are the most time- and money-consuming processes.

The recently finished IARPA Babel program focused on fast development of ASR systems, while the amount of per-language data was decreasing from year to year. The data from 24 low-resource languages were collected, which led to numerous multilingual experiments.

For humans, borrowing the information from other sources when learning a new language is very natural. We all share the same vocal tract architecture and phonetic systems of languages overlap, therefore automatic systems should be able to have the universal and language-independent low-level components (feature extraction and partially also acoustic models), that would be built with various sources of data. In the past, we have verified that the multilingual pre-training is an important technique for feature extraction, especially if limited amount of training data is available [1, 2], similarly to [3, 4]. We have also performed an analysis of combining semi-supervised and multi-lingual training of NN-based bottleneck feature extractors [5]. The hybrid Deep Neural Networks-Hidden Markov Models (DNN-HMM) systems benefit from the multi-lingual training too [6]. Recently in [7], we extended this idea to Bi-directional Long-Short Term Memory Recurrent Neural Networks (BLSTM-RNN) acoustics models and show significant effect of the multilingual pre-training for low resource languages.

The amount of training data over the languages in the Babel project is more or less consistent (50-80h) and limited. Therefore, more detailed analysis of multilingual techniques is not possible. In this paper, we added also English Switchboard and Fisher corpora to investigate BLSTM acoustic models for larger amounts of training data.

2. DATA
The IARPA Babel program data simulate a situation, in which the data for a new language are collected in a limited time. The data consists mainly of conversational telephone speech (CTS) but some scripted recordings and far field recordings are present too. During the 4-year project, datasets of 25 languages were created: Year 1: Cantonese (CA), Pashto (PA), Turkish (TU), Tagalog (TA), Vietnamese (VI). Year 2: Asamese (AS), Bengali (BE), Haitian Creole (HA), Lao (LA), Zulu (ZU), Tamil (Tam). Year 3: Kurdish (KU), Cebuano (CE), Kazakh (KA), Telugu (TE), Lithuanian (LI), TokPisin (TP), Swahili (SW). Year 4: Pashto progress set (about 40h subset of Year 1) (PA2), Javanese (JA), Igbo (IG), Mongolian (MO), Dholuo (DH), Guarani (GU), Amharic (AM), Georgian (not used in this work) (GE). Non-Babel - English used in this work includes: Switchboard-1 Release 2 (SWB), Fisher English Training Speech Part 1+2 (FSH).
### 3. SYSTEM DESCRIPTION

Our systems were built with several toolkits: We used STK/HTK [9] toolkit\(^1\) Kaldi [10] for maximum likelihood (ML) Gaussian mixture model (GMM) training. Finally, we trained BLSTM networks using CNTK [11].

For sake of simplicity, all the results in this work are coming from cross-entropy trained systems with no further use of sequence discriminative criteria, (for example sMBR).

#### 3.1. GMM system

First, GMM based acoustic models are trained to produce phoneme alignments as the labels for the following NN training. These models are based on cross-word tied-states trained from scratch using standard ML algorithm. The baseline GMM systems had approximately 4000 cross-word triphone tied states for Babel and 9100 for full SWB. They were trained on multilingual Region Dependent Transform features trained on 17 Babel languages (Y1-3) as the alignments were found to lead to better NN performance over alignments coming from PLP/MFCC based systems [12].\(^2\) The initial GMM results can be found in Table 2 for Babel languages and in Table 3 for various subsets of SWB.

#### 3.2. BLSTM systems

The BLSTM acoustic-models were trained with last layer producing posterior probabilities of tied-states for HMM models. The latency-controlled BLSTM architecture [13] contains 3 bi-directional layers, for each direction there are 512 memory units and 300 dimensional projection layer. The training is done with truncated back-propagation through time (BPTT) algorithm [14]. Each update is based on \(T_{bptt} = 20\) time-steps of recurrent forward-propagations and back-propagations. For detailed description of the procedure used in our training see [13].

#### 3.3. Feature extraction

The BLSTM NN input is fed with filter bank based features. It contains of 24 log-Mel-filter-bank features concatenated with different pitch features: “BUT F0” has 2 coefficients (F0 and probability of voicing), “snack F0” is a single F0 estimate and “Kaldi F0” has 3 coefficients (F0 normalized with a sliding window, probability of voicing and F0 delta). Fundamental frequency variation (FFV) produces a 7 dimensional vector. The whole feature vector has 24+2+1+3+7=37 coefficients

---

\(^1\) STK is BUT’s variant of HTK: \url{http://speech.fit.vutbr.cz/software/hmm-toolkit-stk}

\(^2\) The scripts for MultRDT features generation can be found in \url{http://speech.fit.vutbr.cz/software/hmm-toolkit-stk}
(see [15] for details on pitch features). These features will be called “FBANK_F0”.

After a conversation-side mean subtraction, we apply a Hamming window and Discrete cosine transform to the feature trajectories spanning 11 frames. We retain 0th to 5th DCT coefficients for each of the original 37 features resulting in 37×6=222 coefficients. These features will be called “11FBANK_F0”.

### 4. MULTI-LINGUAL EXPERIMENTS

#### 4.1. Multilingual architecture

All multilingual models in this work were trained with a ‘block-softmax’ output layer, which consists of per-language softmaxes [16]. The training targets were the context-independent phoneme states, otherwise the size of the final layer would be excessively large.

The NNs were trained with standard cross-entropy objective function and Stochastic Gradient Descent (SGD) approach. Whenever objective degraded on cross-validation data, the learning rate was halved and the previous (so far best performing model) was loaded.

The procedure of porting multilingual models into target language can be described in the following steps:

1. The final multilingual layer (context-independent phoneme states for all languages) is stripped and replaced with a layer specific to target-language (tied-state triphones) with random initialization.
2. This new layer is trained for 8 epochs with a standard learning rate, while the rest of the NN is fixed.
3. Finally, the whole NN is fine-tuned with 10 epochs, the initial value of learning-rate schedule is set to 0.5 of the original value.

#### 4.2. Analysis of feature extraction

Here, we were interested in optimal feature extraction for Multilingual and Monolingual BLSTM architectures. In our recent work [7], we have shown significant gain from using 11FBANK_F0 over bottle-neck features in BLSTM systems. As BLSTM can naturally incorporate context information, any feature stacking might not be necessary, so we experimented with FBANK_F0 features as well.

### Table 4. Comparison of %WER with various initialization and feature extraction.

<table>
<thead>
<tr>
<th>Features</th>
<th>Mult-NN</th>
<th>Javanese</th>
<th>Amharic</th>
<th>Pashto</th>
</tr>
</thead>
<tbody>
<tr>
<td>11FBANK_F0</td>
<td>None</td>
<td>54.4</td>
<td>44.0</td>
<td>50.7</td>
</tr>
<tr>
<td>11FBANK_F0</td>
<td>24L</td>
<td>49.2</td>
<td>39.8</td>
<td>46.1</td>
</tr>
<tr>
<td>FBANK_F0</td>
<td>None</td>
<td>54.0</td>
<td>44.0</td>
<td>49.0</td>
</tr>
<tr>
<td>FBANK_F0</td>
<td>24L</td>
<td>52.1</td>
<td>42.2</td>
<td>47.7</td>
</tr>
</tbody>
</table>

Table 4. Comparison of %WER with various initialization and feature extraction.

#### 4.3. Analysis of number of training epochs

The final multilingual NN is used for the pre-training of a final language specific system, therefore training into “ultimate” minima of the objective function could not be optimal. The multilingual NN has to be able to change its parameters to different languages, therefore early stopping should be taken into account.

For this experiments, the Y1-3 (17L) and Y1-4 (24L) languages were chosen. The first language set simulates adaptation of a multilingual NN to an unknown new language and the second one is showing the case where the target language is contained in multilingual training data.

The first and second columns of Table 5 present significant drop of accuracy when NN is well trained on multilingual data. The first halving of learning rate during the training process was observed on 20th epoch for Y1-3 NN and on 19th for Y1-4 NN. Therefore the final multilingual NN should be taken before this point. Well trained multilingual NN is suitable only in cases where target language is part of multilingual training set, see third and fourth column.

### Table 5. %WER obtained with fine-tuned NNs, which were pre-trained using different number of training epoch.

<table>
<thead>
<tr>
<th>n. epoch</th>
<th>17L Mult-NN</th>
<th>24L Mult-NN</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Javanese</td>
<td>Amharic</td>
</tr>
<tr>
<td>5</td>
<td>50.8</td>
<td>41.2</td>
</tr>
<tr>
<td>10</td>
<td>50.4</td>
<td>40.6</td>
</tr>
<tr>
<td>15</td>
<td>50.1</td>
<td>40.3</td>
</tr>
<tr>
<td>20</td>
<td>50.5</td>
<td>40.4</td>
</tr>
<tr>
<td>25</td>
<td>50.5</td>
<td>40.5</td>
</tr>
<tr>
<td>30</td>
<td>50.9</td>
<td>40.6</td>
</tr>
</tbody>
</table>

Table 5. %WER obtained with fine-tuned NNs, which were pre-trained using different number of training epoch.

According to Table 4, the plain FBANK_F0 features are the most suitable in monolingual systems, so the models can naturally learn context information. But advantage of multilingual pre-training is partly lost with NN trained on this features (comparing to system pre-trained on 11FBANK_F0 features). This is very interesting outcome, it shows that context information is advantageous for multilingual systems. Therefore, 11FBANK_F0 features were used for all multilingual systems and FBANK_F0 were used for plain monolingual baseline systems.

#### 4.4. Multilingual training data

Next, the amount of multilingual training data was investigated. Table 6 presents positive effect of adding more training data into multilingual training, which is consistent with our previous work on feature extraction [2, 7].

In addition, we are presenting the effect of adding rich resource English data sets (last row and column of the table). It shows that having 11 small resource languages is giving better performance than a lot of training data from single language.
Table 6. Comparison of %WER for BLSTM systems with various multilingual initialization.

<table>
<thead>
<tr>
<th>Data size [h]</th>
<th>Monoling.</th>
<th>Multiling. (24L)</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>35.5</td>
<td>26.0 (-9.5)</td>
</tr>
<tr>
<td>50</td>
<td>24.8</td>
<td>21.2 (-3.6)</td>
</tr>
<tr>
<td>100</td>
<td>22.4</td>
<td>19.6 (-2.8)</td>
</tr>
<tr>
<td>150</td>
<td>20.3</td>
<td>18.9 (-1.4)</td>
</tr>
<tr>
<td>200</td>
<td>18.9</td>
<td>17.9 (-1.0)</td>
</tr>
<tr>
<td>All</td>
<td>18.1</td>
<td>17.1 (-1.0)</td>
</tr>
</tbody>
</table>

Table 7. Comparison of SWB %WER on various data size and type of training.

Table 8. Comparison of SWB %WER on various data size and number of layers.

<table>
<thead>
<tr>
<th>Data Size [h]</th>
<th>3L</th>
<th>4L</th>
<th>5L</th>
<th>6L</th>
<th>7L</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>33.8</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>50</td>
<td>23.9</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>100</td>
<td>21.5</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>150</td>
<td>19.5</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>200</td>
<td>18.6</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>All</td>
<td>17.1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 9. %WER for various number of layers on Babel data.

Table 10. %WER for multilingual 6 Layer Babel system.

4.5. Multilingual pre-training on SWB

Next, we were interested in the improvement brought by multilingual pre-training as function of the size of target language training data. Table 7 shows huge improvement for low-amount of training data (10h). After 100h, the gain is starting to saturate. Interestingly, the improvement never completely disappears and even for full SWB, it is still 1% WER absolute. Therefore, multilingual pre-training can play significant role even for rich-resource task.

4.6. Large NNs

Another possible improvement brought by multilingual training is a possibility to pre-train larger NNs that it would be possible on small target language data. We were experimenting with increasing the amount of layers for various SWB data sizes. Surprisingly, monolingual experiments in Table 8 show that 5 layers are better than 3 layers even for 10h of training data. When more than 100 hours of training data is used, 6 layers are giving the best performance.

Similarly to SWB, 5 to 6 layers are more suitable even for Babel languages, see Table 9. Therefore, we decided to retrain multilingual NN on 24 languages with 6 layers as well. The results are in Table 10.

5. CONCLUSION

This paper analyzes multi-lingual training of BLSTM systems. We have shown clear advantage of multi-lingual training of acoustic models in low-resource scenarios. Small but consistent gains are also present on rich resources scenario.

With multilingual pre-training, we have found essential to include context information into multilingual systems even for BLSTM which can naturally learn it.

The optimum size of acoustic model NN was also investigated and we found that even low resource systems (10h) can be trained with 5 layers. The rich resource language can advantageously exploit a more complex system, therefore we are presenting additional gain from building multilingual system on 6 layers.
6. REFERENCES


