LATTICE-FREE MMI ADAPTATION OF SELF-SUPERVISED PRETRAINED ACOUSTIC MODELS

Apoorv Vyas        Srikanth Madikeri        Hervé Bourlard
Idiap-RR-40-2020

DECEMBER 2020
LATTICE-FREE MMI ADAPTATION OF SELF-SUPERVISED PRETRAINED ACOUSTIC MODELS

Apoorv Vyas\textsuperscript{1,2}, Srikanth Madikeri\textsuperscript{1}, Hervé Bourlard\textsuperscript{1,2}

\textsuperscript{1}Idiap Research Institute, Martigny, Switzerland
\textsuperscript{2}Ecole Polytechnique Fédérale de Lausanne, Switzerland
\{avyas, msrikanth, bourlard\}@idiap.ch

ABSTRACT

In this work, we propose lattice-free MMI (LFMMI) for supervised adaptation of self-supervised pretrained acoustic model. We pretrain a Transformer model on thousand hours of untranscribed Librispeech data followed by supervised adaptation with LFMMI on three different datasets. Our results show that fine-tuning with LFMMI, we consistently obtain relative WER improvements of 10\% and 35.3\% on the clean and other test sets of Librispeech (100h), 10.8\% on Switchboard (300h), and 4.3\% on Swahili (38h) and 4.4\% on Tagalog (84h) compared to the baseline trained only with supervised data.

Index Terms— self-supervised pretraining, lfmmi, cross-lingual adaptation, automatic speech recognition

1. INTRODUCTION

A typical approach to exploit unsupervised data for automatic speech recognition (ASR) is to train a seed model using supervised data and use the seed model to automatically transcribe the unsupervised data \cite{1, 2, 3, 4, 5, 6, 7, 8, 9}. This approach is referred to as semi-supervised learning. More recently, self-supervised learning methods that transform the input signal to learn powerful representations have been receiving a lot of attention \cite{10, 11}. In contrast to semi-supervised learning, self-supervised learning aims to improve the seed model by exploiting unlabelled data before adaptation on supervised data.

Semi-supervised learning and self-supervised learning are complimentary as using the self-supervised pretrained network as the staring point for supervised adaptation can potentially improve the seed model which will in-turn improve the quality of transcriptions generated for the unlabelled data. Thus, in this work we focus on adapting a self-supervised pretrained network to get an improved seed model.

Self-supervised training approaches can be broadly grouped into two classes: (1) auto-regressive models that try to predict the future representations conditional on the past inputs \cite{13, 11} and (2) bidirectional models that learn to predict masked parts of the input \cite{10, 15, 12}. In \cite{12, 10}, authors explore the adaptation of a bidirectional network for ASR. However, they only consider cross-entropy based HMM-DNN systems or Connectionist Temporal Classification \cite{16} for supervised training. Moreover, they evaluate the supervised adaptation when unlabelled data and transcribed data belong to the same domains.

The contribution of this work is twofold. First, we show that for the same pretraining method, using LFMMI as the sequence discriminative criterion for adaptation yields a better performance than cross-entropy training. Second, we show that the pretraining remains useful even for out-of-domain supervised data and cross-lingual adaptation.

Specifically, we use the self-supervised pretraining method proposed in \cite{12} to train on thousand hours of librispeech un-transcribed data. We then adapt the pretrained model using LFMMI on three datasets. First, we evaluate on a hundred hour subset of Librispeech. Next, we train on the three hundred hours of Switchboard data, which is conversational speech. Finally, we also evaluate on two low resource languages, Tagalog and Swahili, from the Babel dataset. We always use flat-start LFMMI \cite{17} training for ASR.

The rest of the paper is organized as follows: In Section 2, we describe the details of self-supervised pretraining such as acoustic features, input perturbations, and model architectures. In Section 3, we present the details of the data preparation and supervised training for automatic speech recognition. Finally, in Section 3.2, we present the results on Librispeech, Switchboard, and Babel.

2. SELF-SUPERVISED PRETRAINING

We use the Masked Acoustic Modeling task as described in \cite{12} for self-supervised pretraining of a bidirectional Transformer \cite{18} model. The input to the network is a sequence of acoustic features with a percentage of the input frames masked or noise corrupted. The model attempts to reconstruct the original input given the corrupted input and is trained with $L_1$ loss.

We use the publicly available Librispeech \cite{19} dataset that comprises a total of 960 hours of read speech data. It is di-
vided into three parts: train-clean-100 (100h), train-clean-360 (360h), and train-other-500 (500h). In contrast to [12], we only consider 80 dimensional filterbank energy features for pretraining. We do not use iMLLR features as we need alignments to extract those, which makes them unsuitable in general unsupervised settings.

In the following, we briefly describe the input perturbations used for the Masked Acoustic Modeling task and the model architecture along with the training details.

2.1. Input Perturbations

As described in [12], we apply the following three perturbations to the input acoustic features.

First, we apply time alterations, where we randomly select $T_{num}$ starting points and then mask $7$ consecutive frames. Time alteration blocks can overlap each other resulting in altered block with more than $7$ frames. The selected frames are set to zero with a probability of $0.8$, or replaced with random segments of frame with a probability of $0.1$. For $10\%$ of the times, they are left unaltered. $T_{num}$ is selected such that the total number of masked frames are roughly $15\%$.

We then apply frequency alteration where we randomly mask a block of consecutive channels to zero for all time steps across the input sequence. The number of channels to be masked is selected in the range of $0$ to $W_c$ with equal probability. We set $W_c$ to $16$ in our experiments.

Finally, we apply magnitude alteration where with a probability of $0.15$, we add a noise matrix of the same shape as acoustic features to our input. Each element to the noise matrix is sampled from a Gaussian distribution with mean zero and variance $0.2$.

2.2. Model Architecture

We train two Transformer architectures namely: Tr-med and Tr-small. Tr-med is composed of $12$ encoder layers each with $6$ attention heads. The embedding dimension is set to $64$ for each head and feed-forward dimension is set to $1536$. We use all $960$ hours of data to pretrain this model.

Tr-small has $3$ encoder layers; each with $12$ attention heads. We set embedding dimension to $64$ for each head and feed-forward dimension to $3072$. We only use train-clean-100 subset to pretrain Tr-small.

In our experiments, we compare against the pretrained model presented in [12] referred to as Tr-large, which has $12$ encoder layers. Each layer has $12$ attention heads and uses an embedding dimension of $64$ for each head. The feed-forward dimension is set to $3072$. Tr-large is also trained with all $960$ hours of data. To handle long sequences, we use the improved-clustered attention with $200$ clusters variant proposed in [20].

2.3. Training

Both transformer models are trained with the Adam optimizer [21] with a mini-batch size of $36$. The learning rate is warmed up over the first $7\%$ of total training steps $T_{steps}$ to a peak value of $0.0002$ and then linearly decayed. We set $T_{steps}$ to $200 000$.

3. EXPERIMENTS

We evaluate the pretrained models in two settings. In the first setting, we freeze the weights of pretrained model and use it as a feature extractor. We pass the output of the encoder to a twelve layered factorized time-delay neural network (TDNNF) architecture referred to as TDNNF-large. In the second setting, we fine-tune the pretrained model together with a seven layered TDNNF architecture which we call TDNNF-small. We denote fine-tuning as FT in later experiments. For baseline, we train from scratch TDNNF-large using the same $80$ dimensional filter bank features which were used for pretraining. For TDNNF models, we set hidden layer dimension to $1024$ and bottleneck dimension to $128$.

For supervised training, we use full biphones to enable flat-start training [17] with lattice free maximum mutual information (LMFMI). This enables end-to-end training without the need of prior alignments. We refer to it as e2e-lfmmi in later experiments. Unless specified, we use this for adaptation or training from scratch. All our models are trained in PyTorch [22] using the PkWrap toolkit [23] $^1$.

We apply speed and volume perturbation to increase the dataset to three times. All our models are trained for $15$ epochs with a batch size of $32$. We use Adam [21] optimizer with a learning rate that is decayed from $0.001$ to $0.00003$ using a polynomial decay. When fine-tuning the pretrained model, we set the learning rate for the pretrained network to be $0.00003$ and use the same learning rate policy for the TDNNF network.

3.1. Datasets

We evaluated the performance of the pretrained model on three different datasets with increasing order of difficulties. The first dataset we consider is the $100$ hour subset of Librispeech [19] called train-clean-100. This is the easiest setting as there is no domain shift with respect to the pretraining data. The next dataset we used is the Switchboard [24] with $300$ hours of supervision data. For both Switchboard and Librispeech (pretraining data) the spoken language is English. However Switchboard has conversational speech while Librispeech is read speech. We also present results on two of the Babel [25] languages: Tagalog (84.h) and Swahili (38.5h).

$^1$We will open-source our training scripts and pretrained models.
Table 1. Comparison of word error rates (WER) (in %) on the clean and other parts of the Librispeech test set with and without 4-gram language model rescoring. Fine-tuning the pretrained model with LFMMI significantly outperforms other baselines. Tr-small, Tr-med, Tr-large refers to small, medium, and large Transformers used for pretraining. TDNNF-small and TDNNF-large refer to the 7 and 12 layered TDNNF architectures used for fine-tuning. liGRU refers to light Gated Recurrent Unit. e2e-lfmmi refers to flat-start training with LFMMI and hybrid refers to cross-entropy based HMM-DNN models.

<table>
<thead>
<tr>
<th>Architecture</th>
<th>Features</th>
<th>Pretrain</th>
<th>Supervision</th>
<th>Fine-tune</th>
<th>clean</th>
<th>other</th>
<th>clean</th>
<th>other</th>
</tr>
</thead>
<tbody>
<tr>
<td>3-gram</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4-gram</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Pretraining + Supervised</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(a) Tr-small + TDNNF-large</td>
<td>FBANK</td>
<td>100h</td>
<td>e2e-lfmmi</td>
<td>No</td>
<td>8.91</td>
<td>25.33</td>
<td>6.09</td>
<td>18.80</td>
</tr>
<tr>
<td>(b) Tr-small + liGRU [12]</td>
<td>FBANK</td>
<td>100h</td>
<td>hybrid</td>
<td>No</td>
<td>11.83</td>
<td>NA</td>
<td>9.43</td>
<td>NA</td>
</tr>
<tr>
<td>(c) Tr-med + TDNNF-large</td>
<td>FBANK</td>
<td>960h</td>
<td>e2e-lfmmi</td>
<td>No</td>
<td>7.98</td>
<td>22.14</td>
<td>5.52</td>
<td>16.32</td>
</tr>
<tr>
<td>(d) Tr-med + TDNNF-small</td>
<td>FBANK</td>
<td>960h</td>
<td>e2e-lfmmi</td>
<td>Yes</td>
<td>7.78</td>
<td>20.19</td>
<td>5.35</td>
<td>14.75</td>
</tr>
<tr>
<td>(e) Tr-large + liGRU [12]</td>
<td>fMLLR</td>
<td>960h</td>
<td>hybrid</td>
<td>Yes</td>
<td>8.23</td>
<td>NA</td>
<td>5.84</td>
<td>NA</td>
</tr>
<tr>
<td>Supervised Only</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>8.64</td>
<td>26.27</td>
<td>5.89</td>
<td>20.02</td>
</tr>
</tbody>
</table>

In this experiment, we consider as there is both language and acoustic conditions mismatch.

3.2. Results

In the following we compare the Word Error Rate (WER) achieved by supervised adaptation of pretrained models to the models trained from scratch.

3.2.1. Librispeech (100 hours)

In this experiment, we discuss the setting when the pretrained data and labelled data for ASR come from the same domain. We present our main results in the Table 1. For decoding, we use the model that achieves lowest WER on the dev-clean set. We first decode using a trigram language model and then rescore using a 4-gram language model.

In rows (a) and (b), we compare the effect of training criterion used for adaptation of the pretrained model. It can be seen that using LFMMI for adaptation outperforms hybrid models when FBANK features are used for self-supervised pretraining. Moreover, compared to training from scratch using only the labelled data (f), using Tr-small (a) performs slightly worse on the clean data. This is expected because Tr-small was pretrained using the same 100 hour subset which is used for training (f). Thus both models are exposed to exactly the same amount of data during training. Interestingly (a) performs better than (f) on the other portion of the test data.

In rows (c) and (d), we present the results of pretraining using entire 960 hours of data with Tr-med. We can see that fine-tuning the pretrained model leads to a better performance than simply using it as a feature extractor. Both (c) and (d) outperform the pretrained model with fMLLR features (e) as well as the training only with labelled data (f). Note that we get 6% absolute improvements on the other portion of the test set when fine-tuning with LFMMI. We hypothesize that we get such improvements on the other portion because the pretrained model sees that part of the dataset during the self-supervised pretraining.

From Table 1, it can be concluded using LFMMI for supervision not only results in better performance but also reduces the sensitivity to the features used.

3.2.2. Switchboard (300 hours)

In this experiment, we explore the case when the pretrained data and labelled data for ASR belong to the same language but are different with respect to content, speakers, and acoustic conditions. Switchboard has conversational speech

Table 2. Comparison of word error rates (WER) (in %) on eval2000 test set for the 300 hours Switchboard task. The 3-gram language model is based on Switchboard, whereas the 4-gram employs Switchboard+Fisher training set transcripts. Using pretrained model as a feature extractor or fine-tuning improves the performance over the baseline. Fine-tuning achieves the best WER on both callhome and switchboard part of eval set.

| Model | Hub5’00 (eval2000) | | |
|-------|-------------------|---|---|---|---|
|       | 3-gram            | SWBD | CH | 4-gram | SWBD | CH |
| Pretraining + Supervised | | | | | | |
| Tr-med + TDNNF-large | 11.3 | 22.0 | 9.9 | 19.7 |
| Tr-med + TDNNF-small (FT) | | | | | | |
| Supervised Only | | | | | | |
| TDNNF-large | 11.8 | 22.5 | 10.3 | 20.3 |
| TDNN-LSTM [17] | 11.3 | 21.5 | 9.8 | 19.3 |
recorded at 8 KHz while Librispeech has read speech at 16 KHz. To be compatible with the pretrained models, we resample the Switchboard recordings at 16 KHz before extracting the features. For the TDNNF-large baseline trained only with labelled data, we use the 8 KHz recordings.

Table 2 compares the WER for the models trained from scratch to those that make use of Librispeech pretraining data. In both cases, the pretrained models outperform the model trained from scratch. Once again, the fine-tuned model results in most improvements in terms of WER.

3.2.3. Babel: Swahili and Tagalog

In our final experiment, we consider the scenario when the pretraining data and labelled data for ASR do not share the same language or the acoustic conditions. For this task, we consider the two low resource languages from Babel database: Swahili and Tagalog. Similar to Switchboard setup, we resample the recordings at 16 KHz to be compatible with the pretrained model. Once again, for the TDNNF-large model trained from scratch only on the supervised data, we use the 8 KHz recordings.

Due to the lack of a separate evaluation set, we report results on the dev10h development part of both languages. We remove 1000 utterances from the training set to be used as the development set for model selection. We use trigram language model for decoding.

Table 3 compares the WER for the models trained from scratch to those that make use of Librispeech pretraining data. It can be seen that for both Swahili and Tagalog, using pretrained model as feature extractor performs worse than the models trained from scratch. This indicates the representations learnt by the pretrained model on Librispeech data removes some important information that are specific to these languages resulting in worse performance than the baseline. However, on fine-tuning, the model adjusts its parameters to recapture this information and outperforms the baseline.

<table>
<thead>
<tr>
<th>Model</th>
<th>Swahili</th>
<th>Tagalog</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pretraining + Supervised</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Tr-med + TDNNF-large</td>
<td>40.4</td>
<td>46.6</td>
</tr>
<tr>
<td>Tr-med + TDNNF-small(FT)</td>
<td><strong>36.7</strong></td>
<td><strong>43.4</strong></td>
</tr>
<tr>
<td>Supervised Only</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TDNNF-large</td>
<td>38.3</td>
<td>45.3</td>
</tr>
<tr>
<td>Baselines from other work</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Multi-10 [26]</td>
<td>41.6</td>
<td>46.2</td>
</tr>
<tr>
<td>BLSTM-HMM [26]</td>
<td>38.3</td>
<td>46.3</td>
</tr>
</tbody>
</table>

Table 3. Comparison of word error rates (WER) (in %) on dev10h set for the Swahili and Tagalog languages of the Babel dataset. Fine-tuning the pretrained model with Librispeech data significantly outperforms the monolingual and other reported baselines.

4. CONCLUSIONS AND FUTURE WORK

We proposed lattice-free MMI for supervised adaption of a self-supervised pretrained network for acoustic modeling. We show that fine-tuning a pretrained model with LFMMI criterion outperforms models that are only trained with supervised data resulting in an improved seed model. We further show that pretraining remains useful even under strong distributional shifts.

In future, we intend to combine self-supervision based approaches with multi-lingual training and iterative decoding based semi-supervised training approaches to further improve the performance in low resource settings. We will also fairly compare the performance of other suggested self-supervised training approaches such as wav2vec when the model capacity and pretraining data is fixed.

5. ACKNOWLEDGMENTS

The research leading to these results has received funding from Swiss National Science Foundation project SHISSM (Sparse and hierarchical Structures for Speech Modeling), grant agreement 200021-175589. The research is also partially based upon the work supported by the Office of the Director of National Intelligence (ODNI), Intelligence Advanced Research Projects Activity (IARPA), via AFRL Contract #FA8650-17-C-9116. 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 the ODNI, IARPA, or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright annotation thereon.

6. REFERENCES


Aaron van den Oord, Yazhe Li, and Oriol Vinyals, “Representation learning with contrastive predictive coding,” 2019.


