Improving DNNs Trained With Non-Native Transcriptions Using Knowledge Distillation and Target Interpolation

Amit Das, Mark Hasegawa-Johnson

University of Illinois, USA
amitdas@illinois.edu, jhasegaw@illinois.edu

Abstract

Often, it is quite hard to find native transcribers in under-resourced languages. However, Turkers (crowd workers) available in online marketplaces can serve as valuable alternative resources by providing transcriptions in the target language. Since the Turkers may neither speak nor have any familiarity with the target language, their transcriptions are non-native by nature and are usually filled with incorrect labels. After some post-processing, these transcriptions can be converted to Probabilistic Transcriptions (PT). Conventional Deep Neural Networks (DNN) trained using PTs do not necessarily improve error rates over Gaussian Mixture Models (GMMs) due to the presence of label noise. Previously reported results have demonstrated some success by adopting Multi-Task Learning (MTL) training for PTs. In this study, we report further improvements using Knowledge Distillation (KD) and Target Interpolation (TI) to alleviate transcription errors in PTs. In the KD method, knowledge is transfered from a well-trained multilingual DNN to the target language DNN trained using PTs. In the TI method, the confidences of the labels provided by PTs are modified using the confidences of the target language DNN. Results show an average absolute improvement in phone error rates (PER) by about 1.9% across Swahili, Amharic, Dinka, and Mandarin using each proposed method.

Index Terms: knowledge distillation, target interpolation, deep neural networks, under-resourced, cross-lingual speech recognition

1. Introduction

A well-resourced language (WRL) is a language (e.g. English) with an abundance of resources to support the development of speech technology. Those resources are usually defined in terms of 100+ hours of speech data, corresponding transcriptions, pronunciation dictionaries, and language models. On the contrary, an under-resourced language (URL) lacks one or more of these resources. The most expensive and time consuming resource is the acquisition of transcriptions due to the difficulty in finding native transcribers.

To circumvent this difficulty, transcriptions can be collected from online non-native crowd workers, or Turkers, who neither speak the target language nor have any familiarity with it. Briefly, a single utterance in some target language L is transcribed by multiple Turkers who do not speak L. This generates a collection of non-native transcriptions, one from each Turker. This collection, after merging and some post-processing, can be represented as a confusion network. We refer to such a network as a *Probabilistic Transcription* (PT) [1]. On the contrary, the correct transcription generated by a native speaker can be represented as a single sequence of labels. We refer to this sequence as a *Deterministic Transcription* (DT). DTs are simply conventional transcriptions that we frequently encounter in large vocabulary speech corpora like TIMIT, Wall Street Journal etc.

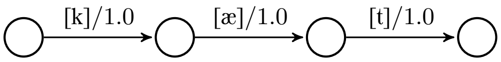

Figure 1: *A deterministic transcription (DT) for the word* cat.

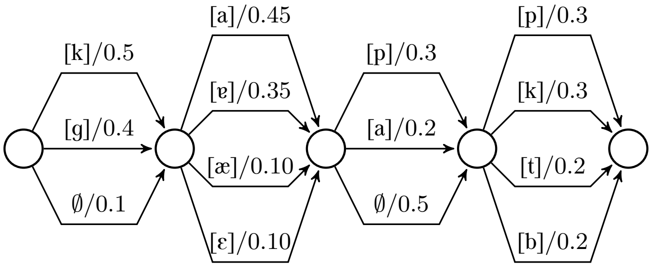

Figure 2: *A probabilistic transcription (PT) for the word* cat.

As an example, consider the DT for the word "cat" in Fig. 1. Each arc represents a label and a probability value which is always one. On the other hand, a PT is the network in Fig. 2. The arc weight specifies the conditional probability that the phoneme was spoken given the audio. The arc weights are determined by agreements among Turker labels. Because Turkers cannot correctly distinguish between all phone pairs in the utterance language, these weights are usually less than 1.0. In terms of training a DNN, running the force alignment using DTs results in 1-hot alignments with each frame associated with only one label. However, force alignment using PTs results in soft alignments since a frame could be associated with multiple labels with non-zero probabilities.

Conventional training of DNNs using PTs do not necessarily improve error rates over GMMs [2, 3]. This is due to higher sensitivity of discriminative training to label noise compared to maximum likelihood training [4]. To alleviate this problem, MTL style training [5], also known as *multilingual training* or *block softmax* [6–8], was introduced as the first reliable baseline to train DNNs using PTs [2]. It uses a mixture of noisy PTs from the target URL and clean DTs from multiple other WRLs [2, 9, 10] in separate sub-tasks. The strong supervision provided by the DTs has the effect of compensating errors in PTs.

In this study, we focus on Knowledge Distillation (KD) and Target Interpolation (TI) to further alleviate the effect of noisy labels in PTs. In [11], the authors describe KD as the process of transfering knowledge from a large cumbersome model (or an ensemble of models) to a small distilled model. The cumbersome and distilled model are sometimes referred to as the Teacher and Student models. Hence, KD is also known as Teacher-Student (TS) learning. If $\mathcal{D}$ is a data set on which the student model is to be trained, then the DNN training procedure involves the following steps. In the first step, feedforward $\mathcal{D}$ through a prior well-trained teacher DNN to generate the posterior outputs (teacher labels). The teacher labels form a soft target distribution for each training example in $\mathcal{D}$. In the second step, train the student DNN by minimizing the cross-entropy (CE) loss between the teacher labels and the posterior outputs of the student DNN. Thus, the student DNN attempts to mimic the behavior of the teacher DNN by trying to match its own outputs with those of the teacher labels. To improve the generalizability of the student DNN, the teacher labels could be generated by using a high temperature T in the softmax of the teacher DNN. The same temperature T is then used at the softmax of the student DNN during CE training. It has been shown that when $T \to \infty$ (high temperature limit), CE training is equivalent to minimizing the mean square error (MSE) of the logits (pre-softmax activations) between the teacher and student DNNs [11].

Several studies [12–22] in the past have used KD to improve DNNs. In [12], a small DNN was trained using teacher labels generated by feedforwarding a large number of untranscribed data through a large DNN. In other studies, the authors transfer the knowledge from a large RNN to a small DNN [13] or from a large DNN to a small highway DNN [14]. In [15,16], KD was used to improve the robustness of DNNs to noisy data. The one that is most relevant to our work is [17] where KD was used for adaptation to under-resourced Japanese dialects.

In the TI approach, we interpolate the confidences of the labels provided by PTs with the confidences of the target language DNN. The DNN is then trained using the new interpolated confidence values. Intuitively, we emphasize the beliefs of the learner rather than solely relying on noisy "ground truth" labels.

The remainder of the paper is organized as follows. In Section 2 and Section 3, we describe the KD and TI frameworks respectively. In Section 4, we discuss our experiments and results. In Section 5, we present our conclusions.

2. Knowledge Distillation (KD)

In this section, we provide a brief outline of the KD framework. Consider an input feature vector $\boldsymbol{x}$. A generalized softmax is a softmax function operating on logits $z_k(\boldsymbol{x})$ and a temperature $T \in \mathbb{R}^+$. Here, $k \in \{1, \cdots, K\}$, where K is the total number of classes. We denote $z_k(\boldsymbol{x})$ as simply z_k and assume the dependence on $\boldsymbol{x}$ is implicit. The output $y_k(T)$ of the generalized softmax is given by,

$$y_k(T) = \frac{\exp\left(z_k/T\right)}{\sum_{j=1}^{K}\exp\left(z_j/T\right)}. \tag{1}$$

There are two extreme cases in Eq. (1). Let $\boldsymbol{y}(T) = [y_1(T) \cdots y_K(T)]'$. For very hot ($T \gg 1$) and cold temperatures ($T \ll 1$), $\boldsymbol{y}(T)$ approaches the uniform and 1-hot distribution respectively. Thus, $\lim_{T\to\infty} y_k(T) = \frac{1}{K}$ and $\lim_{T\to 0} y_k(T) = \mathbb{1}_{[k=\underset{1\leq j\leq K}{\arg\max}\, y_j]}$. The 1-hot distribution is the result of assigning one to the highest element in $\boldsymbol{y}(T)$ while assigning zero to the remaining elements. In the KD framework, the student model is trained to minimize the loss,

$$E_{\text{KD}} = \rho C(\boldsymbol{p}, \boldsymbol{y}(1)) + (1-\rho) C(\boldsymbol{q}(T), \boldsymbol{y}(T)), \tag{2}$$

where,

$$C(\boldsymbol{p}, \boldsymbol{y}(1)) = -\sum_{k=1}^{K} p_k \log y_k(1), \tag{3}$$

$$C(\boldsymbol{q}(T), \boldsymbol{y}(T)) = -\sum_{k=1}^{K} q_k(T) \log y_k(T). \tag{4}$$

The term p_k in Eq. (3) is the posterior probability of label k given the feature vector $\boldsymbol{x}$. Since this is generated from the noisy PTs, it need not be a binary value 0 or 1 as described in Section 1. Thus, $\boldsymbol{p}$ need not be a 1-hot vector. Likewise, $q_k(T)$ in Eq. (4) is the posterior probability of label k generated by feedforwarding $\boldsymbol{x}$ through a teacher DNN equipped with a generalized softmax with temperature T. In other words, it is a teacher label. In the under-resourced scenario, the teacher DNN is a reasonably well-trained multilingual DNN trained with DTs from WRLs. The term $y_k(T)$ in Eq. (4) is the posterior probability of label k generated by feedforwarding $\boldsymbol{x}$ through a student DNN equipped with a generalized softmax with temperature T. The student DNN is the target language DNN trained with PTs from the URL. The term $y_k(1)$ in Eq. (3) is a special case of $y_k(T)$ with $T = 1$. Finally, ρ is a weight that balances the losses in Eq. (3) and Eq. (4).

During backpropagation, the gradient of Eq. (4) with respect to the student logit z_k, i.e., $\frac{\partial C(\boldsymbol{q},\boldsymbol{y})}{\partial z_k}$, is artificially scaled by T^2. This is because the gradient itself is a function of $1/T^2$. Thus, the artificial scaling removes the dependence on T. As a result, the individual backpropagation errors from Eq. (3) and Eq. (4) have similar ranges and can be added meaningfully.

Knowledge distillation specializes to several interesting cases. When $\rho = 1$, Eq. (2) is the same as the standard CE loss. When $0 < \rho < 1$ and $T = 1$, Eq. (2) is equivalent to regularizing the CE loss with Kullback-Leibler Divergence (KLD) [23]. When $\rho = 0$ (indicating the absence of ground truth labels $\boldsymbol{p}$), Eq. (2) can be used for unsupervised adaptation. For example, in the case of $\rho = 0$, $T = 1$ and when the student DNN is not initialized from a teacher DNN, Eq. (2) was used for unsupervised adaptation using the teacher labels obtained from a large teacher DNN [12]. When $\rho = 0$, $T = 1$ and the student DNN is initialized from the teacher DNN, training using Eq. (2) is equivalent to self-training. Here, the teacher labels $\boldsymbol{q}(1)$ are identical to the outputs $\boldsymbol{y}(1)$ of the student DNN before the first weight update of the student DNN. However, after that, the teacher labels are kept constant whereas the student outputs are allowed to differ with every weight update.

3. Target Interpolation (TI)

In this section, we provide a brief outline of the TI framework. We will omit the dependence on T since in this section $T = 1$ always. First, we define $C(f(\boldsymbol{y}), \boldsymbol{y})$ as,

$$C(f(\boldsymbol{y}), \boldsymbol{y}) = -\sum_{k=1}^{K} f(y_k) \log y_k, \tag{5}$$

where $f(.)$ is an element-wise function of $\boldsymbol{y}$ satisfying $f(y_k) \in [0, 1]$ and $\sum_k f(y_k) = 1$. The DNN is trained to minimize the loss,

$$\begin{aligned} E &= \rho C(\boldsymbol{p}, \boldsymbol{y}) + (1-\rho) C(f(\boldsymbol{y}), \boldsymbol{y}), \\ &= C(\rho \boldsymbol{p} + (1-\rho) f(\boldsymbol{y}), \boldsymbol{y}), \end{aligned} \tag{6}$$

where $C(\boldsymbol{p}, \boldsymbol{y})$ is as defined in Eq. (3). The second step in Eq. (6) is due to the linearity of $C(.,.)$ in the first argument. We consider two among several choices of $f(.)$. They are,

$$f(y_k) = \begin{cases} y_k, & \text{(soft)} \\ \mathbb{1}_{[k=\underset{1\leq j\leq K}{\arg\max}\, y_j]}. & \text{(hard)} \end{cases} \tag{7}$$

Plugging in Eq. (7) and Eq. (5) into Eq. (6), we get,

$$E_{\text{soft}} = -\sum_{k=1}^{K} (\rho p_k + (1-\rho) y_k) \log y_k, \tag{8}$$

$$E_{\text{hard}} = -\sum_{k=1}^{K} (\rho p_k + (1-\rho) \mathbb{1}_{[k=\underset{1\le j\le K}{\arg\max}\, y_j]}) \log y_k. \tag{9}$$

Corresponding error gradients for the losses in Eq. (8) and Eq.(9) are,

$$\frac{\partial E_{\text{soft}}}{\partial z_k} = \rho(y_k - p_k) + (1-\rho) y_k (I(y_k) - H(\boldsymbol{y})), \tag{10}$$

$$\frac{\partial E_{\text{hard}}}{\partial z_k} = \rho(y_k - p_k) + (1-\rho)(y_k - \mathbb{1}_{[k=\underset{1\le j\le K}{\arg\max}\, y_j]}), \tag{11}$$

where,

$$I(y_k) = -\log y_k,$$

$$H(\boldsymbol{y}) = -\sum_{k=1}^{K} y_k \log y_k.$$

The motivation behind the choices in Eq. (7) is that we use the label confidences of the DNN $f(y_k)$ to modify the noisy PT labels p_k. Thus, the new ground truth label is an interpolation between p_k and $f(y_k)$. For the soft case, we use the entire output distribution of the DNN. Then the loss in Eq. (8) becomes the standard CE loss with entropy regularization. A DNN trained using this loss function will find a balance between minimizing the CE loss $C(\boldsymbol{p}, \boldsymbol{y})$ while also lowering the entropy of its outputs $C(\boldsymbol{y}, \boldsymbol{y})$. Since PTs are prone to high entropies, lowering the entropies of the DNN outputs is desirable. For the hard case, we simply binarize the DNN outputs to a 1-hot distribution. Compared to the soft case, the hard case ignores the cross-correlations between different classes. In both cases, however, the new interpolated labels still form a valid probability distribution since they sum to one when summed over the K classes.

4. Experiments and Results

4.1. Data

Multilingual audio files were obtained from the Special Broadcasting Service (SBS) network which publishes multilingual radio podcasts in Australia. The corpus is summarized in Table 1. Natively transcribed DTs in Arabic (*arb*), Cantonese (*yue*), and Hungarian (*hun*) were always treated as data from source WRLs. Non-natively transcibed PTs were used as data from the target URL. We experimented with four target URLs - Swahili (*swh*), Amharic (*amh*), Dinka (*din*), and Mandarin (*cmn*) - in a round-robin fashion. For example, if *swh* is the target language, then the training set consists of PTs in *swh* and DTs in the remaining six languages (*amh*, *din*, *cmn*, *arb*, *yue*, *hun*). Thus, the training set excludes DTs in *swh*. In this sense, our experiments fall under the domain of zero-resource speech recognition.

More than 2500 Turkers participated in transcribing, with roughly 30% of them claiming to know only English. The remaining Turkers claimed knowing other languages such as Spanish, French, German, Japanese, and Mandarin. It may be noted that PTs for Mandarin audio were never collected from Mandarin speaking Turkers. The utterances were limited to a length of 5 seconds. This is because the Turkers did not understand the utterance language and it was easier for them to annotate short utterances than long. Since English was the most common language among the Turkers, they were asked to annotate the sounds using English letters. The sequence of letters was not meant to be meaningful English words or sentences since this would be detrimental to the final performance.

Table 1: *SBS Multilingual Corpus.*

Language	Utterances		Phones
	Train	Test	
Swahili (swh)	462	123	48
Amharic (amh)	516	127	37
Dinka (din)	248	53	27
Mandarin (cmn)	467	113	52
Arabic (arb)	468	112	46
Cantonese (yue)	544	148	32
Hungarian (hun)	459	117	65
All	-	-	82

The important criterion was that the annotated letters represent sounds they heard from the utterances as if they were listening to a sequence of nonsense syllables in some exotic language. Since no Turker is likely to generate the perfect transcription, each utterance was transcribed by ten Turkers creating ten different transcriptions per utterance. These transcriptions were converted to phones and merged into a PT [1]. Approximately \$500 was paid per ten Turkers for transcribing an hour of audio. As for DTs, the same set of utterances were transcribed by native speakers in the target language. However, the DTs in the target language were used only for evaluating the ASR performance on the test set.

The training set consists of a) about 40 minutes of PTs in the target URL and, b) about 40 minutes of DTs in multiple WRLs. The development and test sets were worth 10 minutes each. The test utterances were randomly selected to avoid any speaker or gender bias. Going back to our previous example, if *swh* is the target language, then the training set consists of 40 minutes of PTs in *swh* and 40 minutes of DTs each in *amh*, *din*, *cmn*, *arb*, *yue*, *hun* (total $40 \times 6 = 240$).

All experiments were conducted using the Kaldi toolkit [24]. Kaldi source code in C++ and toy examples of the proposed KD and TI frameworks are available in our github repository.[1]

4.2. Experiments

In this section, we describe the features, baseline, and the proposed experiments. Thirteen Mel Frequency Cepstral Coefficients (MFCCs), spliced with +/- 3 neighboring frames, were extracted from speech utterances. These were then transformed using a Linear Discriminant Analysis (LDA) transform followed by Feature-Space Maximum Likelihood Linear Regression (fMLLR) transform resulting in 40-dimensional fMLLR features. These features were kept low dimensional to avoid the curse-of-dimensionality problem which is more likely to occur in under-resourced scenarios. These features were then mean normalized using Cepstral Mean Normalization (CMN) before using them for DNN training.

As for the labels in DNN training, the forced aligned senones obtained from HMM models were treated as the ground truth labels for DNN training. Since a PT is a confusion network, forced alignment performed on a PT produces a training *alignment lattice* as opposed to a conventional training *alignment sequence* from a DT. Running forward-backward recursion on the alignment lattice generates the frame-level posteriors which are soft as opposed to 1-hot. These soft labels were treated as the ground truth labels during DNN training. Phone based language models (LMs) were built from text data in the

[1] `git clone -b teacher-student https://github.com/irrawaddy28/SBS-kaldi-2015`

Table 2: *PERs of different MTL systems trained with CE, KLD, and KD losses. The parameters ρ and T are the weighting and temperature parameters in Eq.* (2).

System	Parameters		Language			
	ρ	T	**swh**	**amh**	**din**	**cmn**
Baseline (CE)	1	1	44.89	60.79	58.65	53.53
KLD	0.6	1	44.11	59.97	58.19	51.00
KLD	0.4	1	44.21	59.36	58.33	50.29
KLD	0.2	1	44.63	59.55	58.65	50.93
KD	0.6	2	44.12	59.82	58.15	50.93
KD	0.4	2	**43.66**	59.40	**57.97**	49.85
KD	0.2	2	44.40	**59.08**	58.26	**49.38**

target language mined from Wikipedia. Consequently, a phone based decoder was used to generate the final ASR hypotheses which were evaluated using PERs. The following experiments were performed in our evaluation:

- **Baseline [2], [10]:** An MTL system was trained consisting of six shared hidden layers and two separate softmax layers (one softmax per task). The shared hidden layers of the MTL system were initialized from a multilingual DNN. Both the tasks were trained to minimize the CE loss. However, the targets at the first softmax were PTs from a target URL. The targets at the second softmax were DTs in the remaining six WRLs. We do not train with DTs in the target URL.
- **KLD Regularization [23]:** Instead of minimizing the standard CE loss, the first task of the MTL system was trained to minimize E_{KD} in Eq. (2) for the special case of $T = 1$ and $0 < \rho < 1$. Specifically, values of $\rho \in \{0.2, 0.4, 0.6, 0.8\}$ were evaluated.
- **Knowledge Distillation:** The first task of the MTL system was trained to minimize E_{KD} in Eq. (2) with $0 < \rho < 1$ and $T > 1$. Specifically, values of $T \in \{2, 3\}$ and $\rho \in \{0.2, 0.4, 0.6, 0.8\}$ were evaluated.
- **Target Interpolation:** The first task of the MTL system was trained to minimize E_{soft} in Eq. (8) or E_{hard} in Eq. (9). Values of $\rho \in \{0.2, 0.4, 0.6, 0.8\}$ were evaluated.

4.3. Results

The PERs comparing the MTL systems trained with CE, KLD, and KD losses are outlined in Table 2. The systems are named eponymously after their loss types. We highlight only the most interesting cases with ρ in the range $0.6 - 0.2$ and $T = 1, 2$. Analyzing each language column in Table 2, it is clear that the KD systems outperform the baseline CE and KLD systems.

Now, we analyze the effect of T and ρ on PERs. Keeping ρ fixed and varying T is equivalent to comparing KLD vs KD systems. Thus, keeping ρ constant, KD systems ($T = 2$) outperform their KLD counterparts ($T = 1$) most of the times. Increasing T makes the class correlations more pronounced. This indicates that the temperature parameter improves the generalization capacity of the DNNs by preventing overfitting to the noisy PTs. Next, keeping $T = 2$ fixed and varying ρ is equivalent to limiting our comparison within the variants of KD systems. As ρ decreases, the PERs tend to decrease first and then increase. Desirable values of ρ are $\rho < 0.5$. From Eq. (2), this implies that the peformance improves when the system relies increasingly on the teacher labels rather than the PT labels. However, this trend reverses for very low values of ρ. For example, in the extreme case when $\rho = 0$ (completely ignoring PT labels), we noticed exceedingly high PERs above 85%. This proves that a combination of PT and teacher labels are more useful than solely using either of them.

Table 3: *PERs of different MTL systems trained with CE and TI losses. The parameter ρ is the weighting parameter in Eq.* (8) *and Eq.* (9).

System	Parameter	Language			
	ρ	**swh**	**amh**	**din**	**cmn**
Baseline (CE)	1.0	44.89	60.79	58.65	53.53
TI (Hard)	0.6	43.96	60.44	58.69	51.14
TI (Hard)	0.4	44.08	59.98	57.94	**49.81**
TI (Hard)	0.2	44.24	60.58	59.19	51.20
TI (Soft)	0.6	43.49	60.19	58.62	51.09
TI (Soft)	0.4	**43.29**	**59.65**	**57.65**	50.02
TI (Soft)	0.2	44.16	61.14	59.26	50.79

Table 4: *Summary of the best proposed systems. Absolute improvements over the baseline system inside parantheses.*

Lang	Baseline (CE)	Best		Parameters	
	PER	**PER**	**System**	ρ	T
swh	44.89	43.29 (1.60)	TI (Soft)	0.4	1
amh	60.79	59.08 (1.71)	KD	0.2	2
din	58.65	57.65 (1.00)	TI (Soft)	0.4	1
cmn	53.53	49.38 (4.15)	KD	0.2	2

The PERs comparing the CE and TI systems are outlined in Table 3. Again, we highlight only the most interesting cases of ρ $(0.6 - 0.2)$. Clearly, both (Hard and Soft) variants of TI systems outperform the baseline CE system. Among the TI systems, TI (Soft) outperforms TI (Hard) for the African languages (Swahili, Amharic, Dinka) whereas it is the opposite for Mandarin. Suprisingly, for both TI (Hard) and TI (Soft), $\rho = 0.4$ is the most desirable value. Quite conveniently, this value of ρ does not change across languages explored in this study. Similar to the KD system, values of $\rho < 0.5$ imply that the peformance improves when the system relies increasingly on the DNN labels rather than the PT labels. This means interpolation is useful and that the new interpolated targets are effective in alleviating the noise in PT labels. However, similar to the KD system, setting $\rho = 0$ results in very high PERs.

Finally, a summary of the best proposed systems for each language, along with their parameters, is highlighted in Table 4. The average improvement is about 1.9% absolute for each KD and TI. This is quite useful considering that this is a zero-resource scenario with no access to reliable ground truth DTs in the target URL.

We conducted additional experiments in an attempt to further boost the performance of the best KD systems. Since the PT distribution $\boldsymbol{p}$ is a soft distribution, we parameterized $\boldsymbol{p}$ with a new temperature parameter T_{PT}. After changing $\boldsymbol{p}$ to $\boldsymbol{p}(T_{\text{PT}})$, we minimze the KD loss E_{KD} in Eq.(2). We noticed an improvement in PER over the best KD systems by about 0.2% absolute when $T_{\text{PT}} = 2$. Since the improvement is marginal, we continue to investigate ways to improve $\boldsymbol{p}$.

5. Conclusions

In this study, we reported further improvements in DNNs trained with noisy non-native transcriptions (PTs) while not having access to native transcriptions (DTs) in the target language. We proposed Knowledge Distillation and Target Interpolation to alleviate the effect of noise in PTs. We observed consistent improvements in PERs for all the languages explored in this study. For each of the proposed methods, we reported an average absolute improvement of 1.9% over the baseline system.

6. References

[1] P. Jyothi and M. Hasegawa-Johnson, "Transcribing Continuous Speech Using Mismatched Crowdsourcing," in *Interspeech*, 2015, pp. 2774–2778.

[2] A. Das and M. Hasegawa-Johnson, "An Investigation on Training Deep Neural Networks Using Probabilistic Transcriptions," in *Interspeech*, 2016, pp. 3858–3862.

[3] A. Das, P. Jyothi, and M. Hasegawa-Johnson, "Automatic Speech Recognition Using Probabilistic Transcriptions in Swahili, Amharic, and Dinka," in *Interspeech*, 2016.

[4] K. Yu, M. Gales, and P. Woodland, "Unsupervised Adaptation with Discriminative Mapping Transforms," vol. 17, no. 4, pp. 714–723, May 2009.

[5] R. Caruana, "Multitask Learning," *Machine Learning*, vol. 28, no. 1, pp. 41–75, Jul 1997.

[6] S. Scanzio, P. Laface, L. Fissore, R. Gemello, and F. Mana, "On the Use of a Multilingual Neural Network Front-End," in *Proc. Interspeech*, 2008, pp. 2711–2714.

[7] K. Veselý, M. Karafiat, F. Grezl, M. Janda, and E. Egorova, "The Language-Independent Bottleneck Features," in *Proc. IEEE SLT*, 2012, pp. 336–341.

[8] J.-T. Huang, J. Li, D. Yu, L. Deng, and Y. Gong, "Cross Language Knowlege Transfer Using Multilingual Deep Neural Network With Shared Hidden Layers," in *Proc. ICASSP*, 2013.

[9] V. H. Do, N. F. Chean, B. P. Lim, and M. Hasegawa-Johnson, "Multi-Task Learning Using Mismatched Transcription for Under-Resourced Speech Recognition," in *Interspeech*, 2017, pp. 2073–2077.

[10] A. Das, M. Hasegawa-Johnson, and K. Veselý, "Deep Autoencoder Based Multi-Task Learning Using Probabilistic Transcriptions," in *Interspeech*, 2017, pp. 2073–2077.

[11] G. Hinton, O. Vinyals, and J. Dean, "Distilling the Knowledge in a Neural Network," in *arXiv:1503.02531*, 2015.

[12] J. Li, R. Zhao, J.-T. Huang, and Y. Gong, "Learning Small-Size DNN with Output-Distribution-Based Criteria," in *Interspeech*, 2014.

[13] W. Chan, N. R. Ke, and I. Lane, "Transfering Knowledge from a RNN to DNN," in *Proc. Interspeech*, 2015, pp. 3264–3268.

[14] L. Lu, M. Guo, and S. Renals, "Knowledge Distillation for Small-Footprint Highway Networks," in *Proc. ICASSP*, 2017, pp. 4820–4824.

[15] K. Markov and T. Matsui, "Robust Speech Recognition Using Generalized Distillation Framework," in *Interspeech*, 2016, pp. 2364–2368.

[16] S. Watanabe, T. Hori, J. L. Roux, and J. Hershey, "Student-Teacher Network Learning with Enhanced Features," in *ICASSP*, 2017.

[17] T. Asami, R. Masumura, Y. Yamaguchi, H. Masataki, and Y. Aono, "Domain Adaptation of DNN Acoustic Models Using Knowledge Distillation," in *Proc. ICASSP*, 2017.

[18] J. Li, M. Seltzer, X. Wang, R. Zhao, and Y. Gong, "Large Scale Domain Adaptation via Teacher-Student Learning," in *Interspeech*, 2017.

[19] J. Cui, B. Kingsbury, B. Ramabhadran, G. Saon, T. Sercu, K. Audhkhasi, A. Sethy, M. Nussbaum-Thom, and A. Rosenberg, "Knowledge Distillation Across Ensembles of Multilingual Models for Low-Resource Languages," in *Proc. ICASSP*, 2017, pp. 4825–4829.

[20] Y. Chebotar and A. Waters, "Distilling Knowledge from Ensembles of Neural Networks for Speech Recognition," in *Proc. Interspeech*, 2016, p. 34393443.

[21] G. Tucker, M. Wu, M. Sun, S. Panchapagesan, G. Fu, and S. Vitaladevuni, "Model Compression Applied to Small-Footprint Keyword Spotting," in *Proc. Interspeech*, 2016, p. 18781882.

[22] J. Li, R. Zhao *et al.*, "Developing Far-Field Speaker System via Teacher-Student Learning," in *Proc. ICASSP*, 2018.

[23] D. Yu, K. Yao, H. Su, G. Li, and F. Seide, "KL-Divergence Regularized Deep Neural Network Adaptation For Improved Large Vocabulary Speech Recognition," in *Proc. ICASSP*, 2013, pp. 7893–7897.

[24] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlíček, Y. Qian, P. Schwarz, J. Silovský, G. Stemmer, and K. Veselý, "The Kaldi Speech Recognition Toolkit," 2011.