HKUST SPD - INSTITUTIONAL REPOSITORY

Title

Authors

Source

Version

DOI

Publisher

Copyright

This version is available at HKUST SPD - Institutional Repository (<https://repository.ust.hk/ir>)

If it is the author's pre-published version, changes introduced as a result of publishing processes such as copy-editing and formatting may not be reflected in this document. For a definitive version of this work, please refer to the published version.

Multi-task Learning of Deep Neural Networks for Low-resource Speech Recognition

Dongpeng Chen and Brian Mak

Abstract—We propose a multi-task learning (MTL) approach to improve low-resource automatic speech recognition using deep neural networks (DNNs) without requiring additional language resources. We first demonstrate that the performance of the phone models of a single low-resource language can be improved by training its grapheme models in parallel under the MTL framework. If multiple low-resource languages are trained together, we investigate learning a set of universal phones (UPS) as an additional task again in the MTL framework to improve the performance of the phone models of all the involved languages. In both cases, the heuristic guideline is to select a task that may exploit extra information from the training data of the primary task(s). In the first method, the extra information is the phoneme-to-grapheme mappings, whereas in the second method, the UPS helps to implicitly map the phonemes of the multiple languages among each other. In a series of experiments using three low-resource South African languages in the Lwazi corpus, the proposed MTL methods obtain significant word recognition gains when compared with single-task learning (STL) of the corresponding DNNs or ROVER that combines results from several STL-trained DNNs.

Index Terms—Multi-task learning, deep neural network, low-resource speech recognition, universal phone set, universal grapheme set.

I. INTRODUCTION

In the research on automatic speech recognition (ASR), huge efforts are spent on the most popular languages such as English, French, German, Mandarin, ... etc. and great success has been achieved. However, due to the lack of audio and language resources, there are still many languages in the world that do not benefit from the advanced human language technologies.

The creation of language resources generally needs the help of native linguistics experts, and is usually costly and time-consuming; it is even more so if only non-native developers are available. Thus, an important research direction in low-resource ASR is to make the process easier and faster. For instance, a semi-automatic approach to preparing a pronunciation dictionary is to first create a small primary dictionary manually, and then extend it to a large dictionary by applying grapheme-to-phoneme conversion [1]. However, the performance of the final dictionary highly depends on the quality of the primary one. A simpler solution is to abandon the phone-based models and employ graphemes as the basic acoustic units because grapheme modeling [2], [3], [4], [5],

[6] does not require a phonetic dictionary¹. Many languages that use an alphabet writing system are suitable for grapheme-based acoustic modeling, and their grapheme set is usually selected to be the same as their set of alphabets.

While ways are sought to create resources for a new language more efficiently, other ways are proposed to reduce the amount of training data required for robust acoustic modeling. For systems based on hidden Markov modeling (HMM), one common solution is parameter tying or sharing [7], [8], [9], [10], [11]. Another method is the basis approach in which a relatively small set of basis vectors or functions is computed so that other model parameters may be derived from them. Successful examples include tied-mixture HMM [12], [13], subspace Gaussian mixture model (SGMM) [14], Bayesian sensing HMM [15], and canonical state model [16]. On the other hand, transfer learning [17] is also proved effective when out of domain data are available. Notable efforts include cross-lingual [18], [19] and multi-lingual [20], [21], [22] acoustic modeling techniques. A basic assumption behind these techniques is that there exist good mappings between phonemes in some rich-resource languages and the phonemes in the low-resource target language so that transfer learning may be applied to transfer the knowledge learned in the former to the latter by adaptation [23], [24], transformation [25], [26], or data augmentation [27]. Usually it is also assumed that these phoneme mappings can be implicitly derived when speeches from multiple languages are trained together and part of their models are shared as in SHL-MDNN [22] or SGMM [28]. [18] explicitly derives the mappings by a data-driven method and a knowledge-based method. In addition, universal acoustic models have also been investigated in which phonemes of multiple languages are all explicitly mapped to a universal phone set (UPS) so that there are sufficient data to train the universal phones [29], [30]. However, it is usually found that the performance of UPS models is worse than the language-specific acoustic models for the target language unless the amount of training data for the target language is really small.

In this paper, we would like to improve the estimation of the phonetic models of a low-resource language by learning other related task(s) together under the *multi-task learning* (MTL) [31] framework using deep neural networks² [32]. According

¹In practice, grapheme models can be trained using existing phone-based ASR softwares. During training, a “pronunciation dictionary” is simulated by simply representing each word by its graphemic transcription.

²Although we mostly take the estimation of phone-based acoustic models as the primary task in the proposed MTL framework, the approach is general and flexible, and one may take other task such as the estimation of grapheme-based acoustic models as the primary task as well.

The work described in this paper was fully supported by grants from the Research Grants Council of Hong Kong SAR, China (Project Nos. 616513 and 16206714). The authors are with the Department of Computer Science and Engineering, the Hong Kong University of Science and Technology, Clear Water Bay, Hong Kong. E-mail: {dpchen, mak}@cse.ust.hk.

to the theory of MTL, *related* tasks can be jointly learned to improve the generalization performance of each task if they share the same inputs and some internal representation; the effect is more prominent when the amount of training data is small. We believe that humans do not learn the sounds of a spoken language in isolation but together with other cues such as their graphemes, the lexical contexts, its similarity with or its differences from other languages, and so forth. Our major contribution in this paper is the successful identification of appropriate related tasks *without* requiring additional resources so that phonetic models of low-resource languages can be better learned under the MTL framework. The heuristic guideline is to find a secondary task related to the primary task that can exploit extra information from its training data. In our case, the extra information being exploited is the implicit mapping between the primary targets and the secondary targets. More specifically, we first propose learning the graphemes of the same language as the secondary task³ in Section III. Grapheme-based acoustic modeling does not require additional language resources besides those already required by phone-based acoustic modeling. We believe that the grapheme learning task exploits extra information in the acoustic training data to learn implicit phoneme-to-grapheme mappings of the language. Then, if several low-resource languages are to be learned together, we propose in our second method in Section IV to derive a UPS among the languages and use the UPS learning as an additional secondary task in the learning of the multi-lingual phonetic models. The UPS learning not only implicitly encodes an *indirect* mapping among the phones of all the involved languages, but also serves as a regularizer for the learning of the phonetic models of each language. Finally we combine the above two methods and further performance gain is obtained.

The rest of this paper is organized as follows. We first review multi-task learning in Section II. The first MTL approach using joint phone-based and grapheme-based acoustic modeling is described in Section III. Section IV describes how multi-lingual acoustic modeling is conducted in an MTL framework by learning universal phone/grapheme models in parallel. Experimental evaluation on three low-resource South African languages is reported in Section V, which is followed by concluding remarks in Section VI.

II. MULTI-TASK LEARNING

Multi-task learning (MTL) [31] or learning to learn [34] is a machine learning approach that aims at improving the generalization performance of a learning task by jointly learning multiple related tasks together. It is found that if the multiple tasks are related and if they share some internal representation, then through learning them together, they are able to transfer knowledge to one another. As a result, the common internal representation thus learned helps the models generalize better for future unseen data. In [35], a statistical learning theory based approach to MTL is developed and a generalization bound on the average error of MTL is derived. In [35], [36], the notion of relatedness among multiple tasks is defined in

a particular way so that a tighter generalization bound for each learning task can be derived. In his thesis [31], Caruana postulated some requirements for related tasks if their joint learning in the MTL approach was to work well:

- (a) related tasks must share input features, and
- (b) related tasks must share hidden units to benefit each other when trained with MTL-backprop.

He also listed out some task relationships which would enable MTL-backprop to learn better internal representation (or more generalized model) of the related tasks: data amplification, eavesdropping, attribute selection, representation bias, and overfitting prevention.

MTL has been applied successfully in many speech, language, image and vision tasks with the use of neural network (NN) because the hidden layers of an NN naturally capture learned knowledge that can be readily transferred or shared across multiple tasks. For example, [37] applies MTL on a single convolutional neural network to produce state-of-the-art performance for several language processing predictions; [38] improves intent classification in goal-oriented human-machine spoken dialog systems which is particularly successful when the amount of labeled training data is limited; in [39], the MTL approach is used to perform multi-label learning in an image annotation application.

A. Multi-task learning in ASR Using DNNs

In ASR, MTL has been applied to improving performance robustness using recurrent neural networks [40]. With the emergence of the recently very successful deep neural network (DNN), one expects that DNNs may be used to further improve MTL performance; we call the resulting deep neural networks *MTL-DNNs*. For instance, Meltzer and Droppo investigated the training of monophone models for TIMIT phoneme recognition together with the learning of the phone labels, state contexts, or phone contexts [41]; significant gains were reported. However, the work did not model triphone states directly and it is not clear if it is really better to use the triphone contexts as the secondary task in learning monophone state posteriors in the MTL framework. MTL has also been employed successfully to train multi-lingual DNNs [21], [22], [42]. In these works, during pre-training and subsequent fine-tuning, data from all training languages are fed through the common hidden layers but each language maintains its own language-specific output layer. However, unlike the MTL work in [31] and [41], for each input only one task is being trained, and the relatedness among the tasks are exploited only by enforcing common weights in the hidden layers. We will follow the notation in [22] and call these multi-lingual DNNs with shared hidden layers SHL-MDNNs.

B. Our MTL-DNN Formulation

We would like to apply the MTL framework to improve phone-based acoustic models for low-resource ASR using DNNs that strictly follows Caruana's two MTL requirements. That is, for each acoustic training input, one or more related secondary tasks are learned in parallel with the primary task of

³The first method has been presented in our conference paper [33].

learning the phonetic models, and the multiple tasks share the hidden layers in a DNN. We believe that the first requirement may give better MTL performance as it allows learning the hidden layers with constructive or destructive error gradients simultaneously from multiple tasks for each input; however, the requirement was not always enforced by some previous works (e.g., SHL-MDNN [22]). In terms of task relationships, we will make use of data amplification, representation bias, and overfitting prevention in the selection of secondary task(s), which, together with the use of early stopping in DNN training, will result in a more generalized acoustic model.

Let's assume there are K tasks $\mathcal{T} \equiv \{\mathcal{T}_1, \mathcal{T}_2, \dots, \mathcal{T}_K\}$ to learn under the MTL framework. The MTL model parameters are represented by $\Lambda \equiv \{\lambda_0\} \cup \{\lambda_1, \lambda_2, \dots, \lambda_K\}$, where λ_0 consists of model parameters that are shared by *all* tasks and λ_k consists of model parameters specific to only task $\mathcal{T}_k$. In our case, λ_0 represents the shared weights from all hidden layers, whereas λ_k represents the weights in each task-specific output layer of $\mathcal{T}_k$. Without loss of generality, $\mathcal{T}_1$ will always be taken as the primary task, and the rest are secondary (or extra) tasks. The training objective function $\mathcal{E}$ is formulated as the weighted sum of the error functions of all the tasks as follows:

$$\mathcal{E}(\mathcal{D}, \Lambda) = \sum_{\mathbf{x} \in \mathcal{D}} \left(\sum_{k=1}^K \beta_k \mathcal{E}_k(\mathbf{x}; \lambda_0, \lambda_k) \right) \quad (1)$$

where $\mathcal{E}_k$ and β_k are the error function and the task weight of $\mathcal{T}_k$ with $\sum_{k=1}^K \beta_k = 1.0$; $\mathcal{D}$ is the whole set of training vectors from all tasks and $\mathbf{x}$ is one input vector. After training, only the model parameters (λ_0 and λ_1) of the primary task $\mathcal{T}_1$ are needed, and those of the secondary task(s) may be discarded.

The key to successful application of MTL is to identify effective related learning tasks in the context of low-resource ASR.

III. METHOD 1: JOINT ACOUSTIC MODELING OF PHONES AND GRAPHEMES WITH A MONO-LINGUAL MTL-DNN

Acoustic modeling of trigraphemes of the *same* language is chosen as the secondary task in the training of its triphone acoustic models using an MTL-DNN for low-resource ASR. That is, the 2 tasks in this MTL are:

- $\mathcal{T}_1$ (primary task): posteriors of triphone senones (tied states)
- $\mathcal{T}_2$ (secondary task): posteriors of trigrapheme senones

There are several motivations for the choice:

- although it may not be absolutely necessary, humans usually learn a language by reading, listening, and speaking. Hence, the joint learning of phonemes and graphemes is a real-life example of MTL, and we would like to repeat its success in ASR;
- past experiences in low-resource ASR [2]–[6] show that when the two tasks are trained individually, they give comparable recognition performance for many languages. Thus, their joint training may benefit each other; and
- grapheme-based acoustic modeling requires no additional language resources besides those already used by phone-

Fig. 1. An MTL-DNN system for joint training of triphone and trigrapheme acoustic models (MTL-DNN-PG). Outputs, labelled as green, from 2 separate tasks are turned “on” by an input acoustic vector.

based acoustic modeling. That is an important advantage in low-resource ASR.

Fig. 1 shows an overview of the proposed MTL-DNN system for the joint training of triphone and trigrapheme acoustic models; we will denote this DNN as MTL-DNN-PG. The DNN architecture is similar to the one used in common multi-lingual ASR [21], [22]. Essentially two single-task learning DNNs (STL-DNNs), one for training triphone models and the other for training trigrapheme models are merged so that their hidden layers are shared, while each of them keeps its own output layer. The two output layers are trained to model the posterior probabilities of triphone senones (tied states) and trigrapheme senones respectively for the *same* given input acoustic frame. That is, given an input acoustic vector $\mathbf{x}$, the posterior probability of the i th triphone senone $s_i^{(1)}$ at the triphone output layer is computed using the following softmax function:

$$P(s_i^{(1)}|\mathbf{x}; \lambda_0, \lambda_1) = \frac{\exp(y_i^{(1)})}{\sum_{i'=1}^{N_1} \exp(y_{i'}^{(1)})}, \quad \forall i = 1, \dots, N_1, \quad (2)$$

where $y_i^{(1)}$ is the activation of the senone $s_i^{(1)}$ and N_1 is the total number of triphone senones (in task $\mathcal{T}_1$). A similar formula may be derived for the posterior probabilities $P(s_i^{(2)}|\mathbf{x}; \lambda_0, \lambda_2)$ of the N_2 trigrapheme senones (in task $\mathcal{T}_2$). For each training frame $\mathbf{x}$, the error function of task $\mathcal{T}_k$ ($k = 1, 2$) is to minimize the following per-frame cross entropy:

$$\mathcal{E}_k(\mathbf{x}; \lambda_0, \lambda_k) = - \sum_{i=1}^{N_k} d_i^{(k)} \log P(s_i^{(k)}|\mathbf{x}; \lambda_0, \lambda_k), \quad (3)$$

where $d_i^{(k)}$ is the target value of the i th senone in $\mathcal{T}_k$. Finally, the task errors are weighted and summed over all training frames as described in Eq. (1).

The triphone and trigrapheme senones in the MTL-DNN-PG are obtained from their corresponding tied-state GMM-HMM systems. The triphone and trigrapheme GMM-HMMs are also utilized to obtain the frame labels and senone priors by

forced aligning the training data. During MTL-DNN training, the target values of exactly one triphone senone output unit *and* one trigrapheme senone output unit will be set to 1.0 per training frame. During decoding, each senone posterior probability is converted back to a scaled likelihood by dividing it by its prior as follows:

$$P(\mathbf{x}|s_i^{(k)}; \lambda_0, \lambda_k) \propto \frac{P(s_i^{(k)}|\mathbf{x}; \lambda_0, \lambda_k)}{P(s_i^{(k)})}. \quad (4)$$

Afterward, Viterbi decoding is performed using either the triphone or the trigrapheme MTL-DNN-HMM⁴. In fact, one may even combine the independent decisions from the triphone decoder and the trigrapheme decoder to get better result using, for example, ROVER [43].

One limitation of using grapheme acoustic modeling as the secondary task is that graphemes may not be the appropriate modeling units for all languages; graphemes are closely related to the acoustic manifestation of alphabetic languages only. Fortunately, the majority of languages in the world is alphabetic [5].

TABLE I
THE UNIVERSAL PHONE SET (UPS) AND THE PHONEMES' USAGE IN THREE SOUTH AFRICAN LANGUAGES.

IPA	UPS	Afr	Ses	siS
Affricates				
t ϕ	tp_b	×	×	✓
t ϕ^h	tS_^	×	✓	×
t ϕ^h	tS_h	×	✓	×
ts'	ts_^	×	✓	✓
ts ^h	ts_h	×	✓	✓
kx	kx	×	✓	×
k ϕ	kK_^	×	×	✓
ps ^h	ps_h	×	✓	×
p ϕ^h	pS_h	×	✓	×
p ϕ'	pS_^	×	✓	×
q ϕ	d_0Z	×	✓	✓
dz	dz	×	×	✓
dB	dB	×	×	✓
Fricatives				
f	f	✓	✓	✓
v	v	✓	×	✓
s	s	✓	✓	✓
z	z	✓	×	✓
ʃ	S	✓	✓	✓
x	x	✓	✓	✓
h	h	×	×	✓
fi	h_b	✓	✓	✓
ɬ	K	×	✓	×
ɬ ϕ	K_b	×	×	✓
fj'	fS	×	✓	×
Clicks				
!	!!_b	×	×	✓
!	!_b	×	✓	×
Diphthongs				
əi	@i	✓	×	×
əy	u_y	✓	×	×
əu	@u	✓	×	×
iə	i@	✓	×	×
uə	u@	✓	×	×
Trills&Flaps				
r	r	✓	✓	✓
—	sil	✓	✓	✓
Stops				
p	p	✓	×	✓
p ^h	p_h	×	✓	✓
p'	p_^	×	✓	×
b	b	✓	✓	×
t	t	✓	×	✓
t ^h	t_h	×	✓	×
t'	t_^	×	✓	×
d	d	✓	×	×
k	k	✓	×	✓
k ^h	k_h	×	✓	✓
k'	k_^	×	✓	×
g	g	✓	×	✓
tl'	tl_^	×	✓	✓
tl ^h	tl_h	×	✓	✓
Nasals				
m	m	✓	✓	✓
n	n	✓	✓	✓
J	J	×	✓	✓
N	N	✓	✓	✓
Vowels				
i	i	✓	✓	✓
y	y	✓	×	×
u	u	✓	✓	✓
I	I	×	✓	×
φ	eu_:	✓	×	×
ε	E	✓	×	✓
ɔ	O	✓	✓	✓
a	a	✓	✓	✓
a:	A:	✓	×	×
ə	@	✓	×	×
æ	{	✓	×	×
ʊ	U	×	✓	×
œ	u_	✓	×	×
Approximants				
l	l	✓	✓	✓
j	j	✓	✓	✓
w	w	✓	✓	✓

⁴Although we start with the goal of improving a phone-based ASR system, as we will see, for some low-resource language, a grapheme-based ASR system may perform better when the amount of training data is small.

Fig. 2. A multi-lingual MTL-DNN system (ML-MTL-DNN-UPS) with shared hidden layers and an extra output layer of UPS states. Outputs, labelled as green, from 2 separate tasks are turned “on” by an input acoustic vector.

IV. METHOD 2: JOINT ACOUSTIC MODELING OF LANGUAGE-SPECIFIC TRIPHONES AND UNIVERSAL PHONES WITH A MULTI-LINGUAL MTL-DNN

When resources from other languages are available, cross-lingual or multi-lingual ASR techniques may be used to improve low-resource language ASR [18]–[29]. In our second method, we propose to improve low-resource acoustic modeling by exploiting the relationship among the phonemes from multiple languages via a universal phone set in the MTL framework without directly defining the mappings between them.

A. Universal Phone Set (UPS)

Many multi-lingual ASR techniques utilize a global phone inventory such as the International Phonetic Alphabet (IPA) [44], or a smaller *universal phone set* (UPS) which can be written in the ASCII format and is derived from the IPA. During multi-lingual acoustic modeling, phones from different languages having the same UPS phonetic symbol will share their training data. In this paper, the three South African languages under investigation come from the Lwazi project [45] which already provides their IPA phoneme sets [46]. Thus, we simply unify their phoneme sets (after removing any duplicates) to form the UPS and write them in the HTK format. Table I shows the final UPS of 67 phonemes and their uses in the three South African languages in our experiments.

B. Multi-lingual MTL-DNN With an Extra UPS Learning Task

To develop a multi-lingual ASR system with K languages, $K + 1$ tasks will be jointly learned in our multi-lingual MTL-DNN system:

- $\mathcal{T}_1$: posteriors of triphone senones of the first language
- $\mathcal{T}_2$: posteriors of triphone senones of the second language
- $\vdots$
- $\mathcal{T}_K$: posteriors of triphone senones of the K th language
- $\mathcal{T}_u$: posteriors of monophone states of the UPS

With respect to the k th language, $\mathcal{T}_k$ is the primary task and the remaining K tasks $\{\mathcal{T}_1, \dots, \mathcal{T}_{k-1}, \mathcal{T}_{k+1}, \dots, \mathcal{T}_K, \mathcal{T}_u\}$ are the secondary tasks with the UPS task $\mathcal{T}_u$ being a common task for the learning of *any* language. The proposed multi-lingual MTL-DNN system is shown in Fig. 2. The model architecture is similar to SHL-MDNN in [21], [22] except for the additional output layer for learning the posteriors of UPS monophone states; we will call our multi-lingual MTL network *ML-MTL-DNN-UPS*. Without the additional UPS task, one simply hopes that the shared hidden layers will automatically capture the phonetic-acoustic relationships among the multiple languages during MTL-DNN training so that the recognition performances of all languages are enhanced. Our additional UPS learning task forces this to happen: the weights in the shared hidden layers are trained to cause acoustic vectors from different languages that are mapped to the same UPS phoneme to activate the same target in the UPS output layer. Moreover, instead of directly defining $O(K^2)$ phoneme mappings between any two of the K languages, one only needs to map the phonemes of each of the K languages to the UPS phonemes.

Let's denote the training data set from the k th language as $\mathcal{D}^{(k)}$ and its set of N_k triphone senones as $s_i^{(k)}$, $i = 1, \dots, N_k$, and $k = 1, \dots, K$. Similarly, the set of N_u UPS monophone states is denoted as $s_i^{(u)}$. For each input vector $\mathbf{x}^{(k)} \in \mathcal{D}^{(k)}$, only two tasks are involved: the triphone senones of the k th language ($\mathcal{T}_k$) and the UPS monophone states ($\mathcal{T}_u$) are activated using the softmax function of Eq. (2). Their corresponding per-frame cross-entropies, $\mathcal{E}_k(\mathbf{x}^{(k)}; \lambda_0, \lambda_k)$ and $\mathcal{E}_u(\mathbf{x}^{(k)}; \lambda_0, \lambda_u)$, where λ_u consists of the weights in the output layer of the UPS states, are given by Eq. (3). Finally, the training objective function over all data of the multiple languages is modified from Eq. (1) as follows:

$$\begin{aligned} \mathcal{E}(\{\mathcal{D}^{(k)}\}, \Lambda) &= \sum_{k=1}^K \sum_{\mathbf{x}^{(k)}} \left(\beta_k \mathcal{E}_k(\mathbf{x}^{(k)}; \lambda_0, \lambda_k) + \beta_u \mathcal{E}_u(\mathbf{x}^{(k)}; \lambda_0, \lambda_u) \right) \\ &= \underbrace{\sum_{k=1}^K \left(\beta_k \sum_{\mathbf{x}^{(k)}} \mathcal{E}_k(\mathbf{x}^{(k)}; \lambda_0, \lambda_k) \right)}_{\text{as in SHL-MDNN}} \\ &\quad + \underbrace{\beta_u \left(\sum_{k=1}^K \sum_{\mathbf{x}^{(k)}} \mathcal{E}_u(\mathbf{x}^{(k)}; \lambda_0, \lambda_u) \right)}_{\text{regularization}}. \end{aligned} \quad (5)$$

Eq. (5) shows that our multi-lingual MTL-DNN training is different from SHL-MDNN training and may be considered as a regularized version of the latter — a form of regularized MTL [47]. If the language task weights are large, it will be the same as SHL-MDNN training; if the UPS task weight is large, it will be reduced to UPS training. Since UPS models are usually not as good as language-specific models [30], the learned UPS output layer will not be used in recognition, and it is only used to help enforce the cross-lingual phoneme mappings during MTL-DNN training. The training procedure

of ML-MTL-DNN-UPS is similar to that of MTL-DNN-PG in Section III.

From the perspective of regularization, we prefer simpler regularizer and thus we use UPS monophone states instead of UPS triphone senones as the common task. In some preliminary experiments, we also empirically found that they gave similar results.

C. Extensions

As said in Section III, grapheme-based acoustic modeling is a viable solution for low-resource language ASR. Method 2 can be easily modified to use graphemes instead of phones as the modeling units. A universal grapheme set (UGS) may again be created by simply taking the union of the grapheme sets of all the languages under investigation. The UGS for the three languages in our experiments consists of 30 graphemes including one that denotes silence. We will call the grapheme-based multi-lingual MTL-DNN with the extra UGS learning task *ML-MTL-DNN-UGS*.

Obviously, one may further combine Method 1 and Method 2 to jointly model multi-lingual phones and graphemes using the UPS and UGS as the extra learning tasks, and we will label such network as *ML-MTL-DNN-UPS-UGS*. If there are K languages to learn simultaneously, then there will be totally $2K + 2$ (softmax) output layers in the model. Each input acoustic vector from the k th language will activate four targets: one triphone senone and one trigrapheme senone of the k th language, one universal (mono)phone state, and one universal (mono)grapheme state.

V. EXPERIMENTAL EVALUATION

The two proposed multi-lingual MTL-DNN training methods were evaluated on three low-resource South African languages in the Lwazi project [45].

A. The Lwazi Speech Corpus

The Lwazi project was set up to develop a telephone-based speech-driven information system in South Africa. In the project, the Lwazi ASR corpus [45] was collected over a telephone channel from approximately 200 native speakers for each of the 11 official languages in South Africa. Each speaker produced approximately 30 utterances, in which 16 of them are phonetically balanced read speech and the remainders are elicited short words such as answers to open questions, answers to yes/no questions, spelt words, dates, and numbers. A 5,000-word pronunciation dictionary was also created for each language, which covers only the most common words in the language. Thus, for the phone-based experiments, the *DictionaryMaker* software [48] was used to generate dictionary entries for the uncovered words in the corpus.

Three languages were selected from the corpus in our evaluations. They are Afrikaans, Sesotho, and siSwati. Afrikaans is a Low Franconian, West Germanic language, originated from Dutch; Sesotho is a Southern Bantu language, closely related to other languages in the Sotho-Tswana language group; SiSwati is also a Southern Bantu language, but is more

TABLE II
NUMBER OF PHONEMES AND GRAPHEMES OF 3 SOUTH AFRICAN
LANGUAGES AND THE TEST-SET PERPLEXITIES OF THEIR LMS.

Language	#Phonemes	#Graphemes	LM Perplexity
Afrikaans	37	31	11.18
Sesotho	41	25	19.69
siSwati	41	25	10.94
Universal set	67	30	-

TABLE III
DETAILS OF VARIOUS LWAZI DATA SETS. OOV MEANS
“OUT-OF-VOCABULARY” AND “-S” MEANS “SMALL TRAINING SET”.

Data Set	#Spkr	#Utt	Dur(hr)	Vocab	OOV
Afrikaans:					
Train-S	160	1,195	0.82	1,159	—
Train	160	4,784	3.37	1,513	—
Dev	20	600	—	870	0.89%
Eval	20	599	—	876	0.97%
Sesotho:					
Train-S	162	1,206	1.43	1,513	—
Train	162	4,826	5.70	2,360	—
Dev	20	600	—	1,096	1.86%
Eval	20	601	—	1,089	2.29%
siSwati:					
Train-S	156	580	1.02	1,833	—
Train	156	4,643	8.38	4,645	—
Dev	20	599	—	1,889	6.14%
Eval	20	596	—	1,851	4.53%

closely related to the Nguni language group. Thus, the three chosen languages come from different language families. The numbers of phonemes and graphemes in the three languages and the size of the corresponding universal phoneme and grapheme sets are shown in Table II. Since the corpus does not define an official training, development, and test set for each language, we followed the partitions used in [6]. In addition, in order to evaluate the efficacy of MTL in the scenarios where acoustic data is scarce, smaller data sets consisting of approximately one hour of speech were further created by randomly sampling from the full training set of each language. Care had been taken to ensure that there are roughly the same number of utterances for each speaker. Details of the various data sets are listed in Table III.

B. Baseline Systems

For each language, HMM-based recognition systems were built using the proposed two MTL-DNN training methods, and they are compared with two kinds of baseline systems: GMM-HMMs and STL-DNN-HMMs.

1) *Training of the GMM-HMM Baselines:* Acoustic models of all phone-based and grapheme-based baseline systems were strictly left-to-right 3-state continuous-density hidden Markov models (HMMs). HMM state emission probabilities were modeled by Gaussian mixture models (GMMs) with at most 16 components. The GMM-HMMs were trained using maximum-likelihood estimation with 39-dimensional acoustic feature vectors extracted at every 10ms over a window of 25ms from the training utterances. The acoustic features consist of the first 13 PLP coefficients, including c0, and their first- and second-order derivatives. Speaker-based cepstral mean subtraction and

variance normalization were applied to the extracted features before they were used. Moreover, states were tied using phonetic decision trees and the optimal number of tied states (senones) were determined using the development data set.

2) *Training of the STL-DNN Baselines:* Single-task learning (STL) DNNs were trained to classify the central frame of each 15-frame acoustic context window. Feature vectors in the window were concatenated and then normalized to have zero mean and unit variance over the whole training set. All DNNs in our experiments are multi-layer perceptrons (MLPs) with 4 hidden layers and 2048 nodes per layer. The weights of the hidden layers were initialized by unsupervised pre-training a deep belief network (DBN) of the same architecture [49]. The DBN was built by stacking layers of restricted Boltzmann machines (RBMs) on top of one another, and the RBMs were trained one layer at a time. During pre-training, the mini-batch size was kept at 128 (input vectors), and a momentum of 0.5 was employed at the beginning which was then grown to 0.9 after 5 iterations. For Gaussian-Bernoulli RBMs, training kept going for 220 epochs with a learning rate of 0.002, while Bernoulli-Bernoulli RBMs were trained for 100 iterations with a learning rate of 0.02. After pre-training, a softmax layer was added on top of the DBN to obtain the final DNN. The DNN is now a typical feed-forward MLP and was trained by standard stochastic gradient descent. The targets were derived from the senones of the respective GMM-HMM baseline models. The whole network was fine-tuned with a learning rate starting at 0.02 which was subsequently halved when performance gain on the validation set was less than 0.5%. Training continued for at least 10 iterations and was stopped when the classification error rate on the development set started to increase.

It should be noted that the same DBN can be used to initialize both the triphone STL-DNN and the trigrapheme STL-DNN; they only differ in their output softmax layer.

C. Decoding

Standard Viterbi algorithm was used for decoding in all experiments using a bigram language model (LM) for each language. Each LM was trained using only the transcriptions in the training set of its language. The test-set perplexities of these LMs are given in Table II. All system parameters such as the grammar factor and insertion penalty were tuned using the development data.

D. Evaluation 1: Joint Acoustic Modeling of Phones and Graphemes of Lwazi Languages by MTL-DNN-PG

For each language, a single MTL-DNN, labelled as *MTL-DNN-PG*, was trained to estimate the posterior probabilities of both triphone and trigrapheme senones of the language.

1) *MTL-DNN Training:* The construction of the MTL-DNN-PG is very similar to that of an STL-DNN. Firstly, the weights in its hidden layers were initialized by the weights of the same DBN of the corresponding STL-DNNs. But now the output layer in the MTL-DNN-PG consists of two separate softmax layers: one for the primary task and one for the secondary task. For each training sample, two error signals —

TABLE IV

LWAZI: WERS (%) OF **MONO-LINGUAL** SYSTEMS TRAINED ON THE FULL TRAINING SETS. FIGURES IN () ARE #SENONES AND FIGURES IN [] ARE WER REDUCTIONS (%) OVER THE PHONETIC GMM-HMM BASELINE.

State Model	Units	Afrikaans	Sesotho	siSwati
GMM	triphones	9.3 (641)	24.4 (741)	20.2 (339)
	trigraphemes	10.6 (728)	24.3 (543)	20.0 (931)
STL-DNN	triphones	7.2 [22.6]	20.1 [17.6]	18.0 [10.9]
	trigraphemes	8.0	20.4	18.2
MTL-DNN-PG	triphones only	6.4 [31.2]	19.5 [20.1]	17.5 [13.4]
	trigraphemes only	7.6	19.8	18.0
ROVER: triphone STL-DNN + trigrapheme STL-DNN		6.7 [28.0]	19.7 [19.3]	17.4 [13.9]
ROVER: MTL-DNN triphones + same MTL-DNN trigraphemes		6.2 [33.3]	19.3 [20.9]	17.0 [15.8]

TABLE V

LWAZI: WERS (%) OF **MONO-LINGUAL** SYSTEMS TRAINED ON ~1-HOUR SMALL TRAINING SETS. FIGURES IN () ARE #SENONES AND FIGURES IN [] ARE WER REDUCTIONS (%) OVER THE PHONETIC GMM-HMM BASELINE.

State Model	Units	Afrikaans	Sesotho	siSwati
GMM	triphones	12.5 (514)	30.0 (722)	27.1 (271)
	trigraphemes	14.5 (210)	27.7 (324)	24.6 (243)
STL-DNN	triphones	9.5 [24.0]	23.1 [23.0]	21.4
	trigraphemes	11.8	23.5	19.8 [26.9]
MTL-DNN-PG	triphones only	8.9 [28.8]	22.1 [26.3]	20.6
	trigraphemes only	11.3	23.1	18.9 [30.3]
ROVER: triphone STL-DNN + trigrapheme STL-DNN		9.2 [26.4]	22.4 [25.3]	19.3 [28.8]
ROVER: MTL-DNN triphones + same MTL-DNN trigraphemes		8.7 [30.4]	21.8 [27.3]	18.4 [32.1]

one from each task’s softmax layer — were propagated back to the hidden layers. Thus, the learning rate of the hidden layers was set to half of the original one, while that of the output layer remains the same. Otherwise, the training procedure was the same as that of STL-DNN. In addition, the task weights were set to 0.5 for both tasks as other values did not make much difference in our preliminary experiments.

2) *Results and Discussions*: The evaluation was first performed using the full training data set of each language, and then repeated with the reduced training sets to investigate the effect of limited amount of training data on MTL. The recognition performance of the MTL-DNN-PGs are compared with the corresponding GMM-HMM baselines, STL-DNN baselines, and the ROVER integration of the triphone and trigrapheme STL-DNNs, as well as the ROVER integration of the triphone models and trigrapheme models derived from the MTL-DNN-PGs, and they are listed in Table IV and Table V. We have the following observations:

- For all the three languages, when the full training data sets were used for acoustic modeling, both triphone and trigrapheme GMM-HMMs give similar recognition performance. Similar findings were reported in [3] and [4] though the latter used larger amounts of training data (8–80 hours) than what are available in the Lwazi corpus (3–8 hours). Among the three languages, the GMM-HMMs perform the best in Afrikaans and the worst in Sesotho

even though the amount of training data is the least in Afrikaans and the highest in siSwati. The results may be partly explained by the highest LM perplexity in Sesotho. Moreover, it probably means that the acoustic manifestations of the phonemes and graphemes in Afrikaans are less confusable.

- When the training data sets were reduced to about an hour, the recognition performance in all three languages drops as expected. However, trigrapheme models start to outperform the triphone models in siSwati and Sesotho. One reason may be that there are much fewer graphemes than phonemes in the two languages: the ratio is 1:1.6 in these two languages but is 1:1.2 in Afrikaans. Thus, the trigrapheme models were better trained than the triphone models with the smaller amount of data. In fact, the better performance disappears when the full training set was used. The finding again supports the use of graphemic acoustic models in low-resource ASR.

TABLE VI

NUMBER OF MODEL PARAMETERS WHEN THE MODELS WERE ESTIMATED USING THE REDUCED DATA SETS (IN MILLIONS).

State Model	Units	Afrikaans	Sesotho	siSwati
GMM	triphones	0.650	0.913	0.343
	trigraphemes	0.265	0.410	0.307
STL-DNN	triphones	14.8	15.3	14.3
	trigraphemes	14.2	14.4	14.3
MTL-DNN-PG		15.3	15.9	14.8
SHL-MDNN	triphones	16.9		
	trigraphemes	15.4		
ML-MTL-DNN-UPS		17.2		
ML-MTL-DNN-UGS		15.6		
ML-MTL-DNN-UPS-UGS		19.0		

- All phone-based and grapheme-based STL-DNN-HMMs outperform their GMM-HMM counterparts by 9–25% relative in the full training sets, and 15–24% relative in the reduced training sets. The amount of performance gains are typical in large-vocabulary ASR (e.g., [50]) and here we show that such gains can also be obtained in low-resource ASR. This is surprising given that the number of model parameters in STL-DNNs is generally much greater than that in GMMs. Table VI shows the number of model parameters⁵ in the various kinds of state models estimated using the reduced training data sets in the three languages. It can be seen that the STL-DNNs are bigger than the GMMs by more than an order of magnitude. We attribute the robust estimation of the large number of DNN parameters to the effective initialization of the DNN weights by the corresponding pre-trained DBN and/or the effective discriminative fine-tuning of the parameters by back-propagation without overfitting them.
- After MTL was applied to jointly training the triphone and trigrapheme posteriors in a single MTL-DNN, compared with the corresponding STL-DNN, word error rates

⁵The figures do not include HMM transition probabilities but only parameters describing HMM state probability distributions.

(WERs) were further reduced by 3–9% absolute in the full set and 3–5% absolute in the reduced set. Consistent performance gain is observed for both the larger and smaller training sets, and in both the primary and secondary tasks. The results show that MTL benefits learning of not only the primary task but also the secondary task, and it is still effective with even an hour of training speech. Furthermore, the gains are obtained with no additional language resources.

- The triphone models derived from the MTL-DNN-PGs even outperform the ROVER integration of the corresponding triphone and trigrapheme STL-DNNs (except for the case of using the reduced set in siSwati where the trigrapheme models derived from the MTL-DNN-PG is better). This shows that knowledge transfer between multiple learning tasks can be done more effectively by MTL than ROVER integration. Nevertheless, ROVER may still take advantage of any complementary residual errors made by the triphone and trigrapheme models derived separately from the MTL-DNN-PGs and gives the best recognition performance by integrating them. At the end, the best results reduce the WERs of the GMM-HMM baselines by 16–33% relative in the full training set and 27–32% relative in the reduced training set.

To see the generalization effect of MTL-DNN-PG training, we look at the frame classification errors over both the reduced training and development data sets after each back-propagation epoch during both STL-DNN training and MTL-DNN training. The results for Sesotho are plotted in Fig. 3; similar behaviors are also found for Afrikaans and siSwati. The plots clearly show that although MTL-DNN-PG training converges to a worse local optimum than STL-DNN training for the training data, it performs better on the unseen development set. Thus, we may conclude that the extra grapheme modeling task really provides a representation bias to a better local optimum which generalizes better for unseen data.

Fig. 3. Frame classification error rates of STL-DNN and MTL-DNN on the Lwazi training and development sets of Sesotho during back-propagation.

TABLE VII

TIMIT: RECOGNITION PERFORMANCE IN TERMS OF PHONE ERROR RATE (PER) AND GRAPHEME ERROR RATE (GER).

State Model	PER (%)	GER (%)
GMM	28.20	42.64
STL-DNN	21.88	38.11
MTL-DNN-PG	21.29	36.70

E. Evaluation 2: TIMIT Phoneme Recognition By MTL-DNN-PG

To further check the efficacy of using grapheme modeling as the secondary task in Method 1, the experiments in Evaluation 1 were repeated to recognize English phonemes in TIMIT [51] — in a language that is notorious for the complicated relationship between its writing and pronunciation. In fact, grapheme-based acoustic models perform much worse than phone-based acoustic models in English [5]. Because of our better understanding of the English language (and we do not understand the South African languages at all), this evaluation is also designed to verify our claim that the proposed MTL-DNN-PG method exploits extra information in the acoustic data — which is the implicit phoneme-to-grapheme mappings — to learn a more generalized acoustic model. The experimental setup and procedure used to build the various models were very similar to the ones in Evaluation 1; only the differences will be described below.

1) *The TIMIT Corpus*: The standard NIST training set consisting of 3,696 utterances from 462 speakers was used for training, whereas the standard core test set consisting of 192 utterances spoken by 24 speakers was used for evaluation. The development set is part of the complete test set, consisting of 192 utterances spoken by 24 speakers. Speakers in the training, development, and test sets do not overlap. We followed the standard experimentation on TIMIT and collapsed the original 61 phonetic labels in the corpus into a set of 48 phones for acoustic modeling; the latter were further collapsed into the standard set of 39 phones for error reporting. Moreover, the glottal stop [q] was ignored.

2) *GMM-HMM Baselines*: In the phone-based system, there were altogether 15,546 cross-word triphone HMMs based on 48 base phones and 587 senones. Phone recognition was performed with a phone bigram LM that was trained only from the TIMIT training transcriptions, and it has a perplexity of 16.44 on the core test set. The grapheme-based system made use of the 26 English alphabets plus the silence symbol as the graphemic labels. Its GMM-HMMs had altogether 760 senones. A grapheme bigram LM was estimated from the training transcriptions, and it has a perplexity of 22.79 on the core test set — which is very high given that there are only 26 letters to recognize!

3) *DNN Systems*: The training procedure of STL-DNN and MTL-DNN systems on TIMIT was identical to that in Evaluation 1 except that the acoustic features were filter-bank outputs instead of PLP coefficients. Moreover, the softmax output layers consisted of monophone and/or monographeme states as it is usually found that the use of context-dependent monophones does not give performance gain in DNN-HMM

systems for TIMIT.

Fig. 4. The relationship matrix between phone weight vectors (abscissa) and grapheme weight vectors (ordinate) in the MTL-DNN-PG trained on TIMIT.

4) *Results and Discussions:* Results on the core test set are summarized in Table VII.

- We may see that English grapheme recognition is far more difficult than English phone recognition with more than 10% higher error rate. This is expected in English when the estimated grapheme bigram has a perplexity of 22.79; that means the LM does not help much in TIMIT grapheme recognition.
- The STL-DNN-HMM system again outperforms the GMM-HMM system by a large margin — 22.4% relative in phone recognition⁶ and 10.6% relative in grapheme recognition.
- Using grapheme modeling as a secondary task in MTL-DNN-PG training again helps improve the English phone models and lowers the PER by 2.70% relative. The PER reduction obtained in TIMIT is similar to the WER reduction obtained in Sesotho and siSwati in Evaluation 1. Thus, we conclude that grapheme modeling can be a good secondary MTL task for training phone models even for languages in which the relationship between graphemes and phonemes is not strong.
- To visualize the relationship between the trained English monophone posteriors and grapheme posteriors, we compute the cosines of the angles between any two weight vectors in the output layers — one from the monophone output layer and one from the monographeme layer. The results are plotted as a relationship matrix in Fig. 4 in gray scale; a darker cell indicates a stronger relationship between the corresponding phoneme and grapheme. The relationships described in the matrix generally agree well with what we expect. For example, according to Fig. 4, the letter ‘c’ is mostly related to the phonemes [k] and [s], while the letter ‘f’ is mostly related to the phonemes [f], [v] and [th], and so forth. The figure provides some evidence that the MTL-DNN-PG encodes the grapheme-to-phoneme mappings in English.

⁶Our DNN baseline result is comparable with others. For example, one Microsoft group recently reported a PER of 21.63% [41] though a stronger baseline of 20.7% was reported by Hinton’s group in [32].

F. Evaluation 3: Does MTL-DNN-PG Work for Larger Tasks?

Although we design the MTL-DNN-PG method with tackling low-resource ASR in mind, there is no reason why it may not be applied to a larger ASR task with adequate training data. Here we further evaluate MTL-DNN-PG on the Wall Street Journal speech recognition task (WSJ0).

1) *Speech Corpus and Experimental Setup:* The standard WSJ0 SI-84 training set with 15 hours of speech was used for acoustic modeling. Evaluation was performed on the standard Nov92 5K non-verbalized test set, and the si_dt_05 data set was used as the development set for tuning system parameters. Evaluation was performed with both bigram and trigram language models.

2) *Acoustic Modeling:* Steps and configurations of feature extraction, GMM-HMM training, and DNN-HMM training are exactly the same as those in the low-resource South African ASR tasks. Triphone GMM-HMMs with 1254 tied-states and 32 Gaussian mixtures per state was found to be optimal, while the best number of tied-states for trigrapheme modeling is 1489. The triphone and trigrapheme tied-states are then used as DNN outputs.

TABLE VIII
WSJ0: WERS (%) OF VARIOUS SYSTEMS. FIGURES IN [] ARE WER REDUCTIONS (%) OVER THE PHONETIC GMM-HMM BASELINE.

State Model	Units	Bigram	Trigram
GMM	triphones	6.7	4.2
	trigraphemes	12.9	9.6
STL-DNN	triphones	5.1 [23.9]	3.2 [23.8]
	trigraphemes	8.6	6.5
MTL-DNN-PG	triphones only	4.5 [32.8]	2.7 [35.7]
	trigraphemes only	7.7	5.8

3) *Results and Discussions:* Table VIII lists the results on the Nov92 test set. Similar to what is observed on the TIMIT phoneme recognition task, even though trigrapheme modeling is far inferior to triphone modeling for English, training the two tasks together using our MTL-DNN-PG method benefits both of them, reducing the WER of triphone STL-DNN by 15.6% (11.8%) with trigram (bigram) language model.

G. Evaluation 4: Multi-lingual Acoustic Modeling with Universal Phone Set and/or Universal Grapheme Set by ML-MTL-DNN-UPS/UGS

Method 2 of MTL-DNN was evaluated and compared with the existing SHL-MDNN on multi-lingual acoustic modeling of the three South African languages tested in Evaluation 1.

1) *MTL-DNN Training:* The training of the various DNNs in Method 2 is similar to that in Method 1 except that now the training data of all the three languages were pooled together to jointly train their acoustic models. Specifically, the following models were trained and compared:

- *ML-STL-DNN:* multi-lingual STL-DNN of universal (mono)phones or (mono)graphemes;
- *SHL-MDNN:* multi-lingual phonetic shared-hidden-layer DNN [22] (with a total of 3 learning tasks);

- *ML-MTL-DNN-UPS*: multi-lingual phonetic MTL-DNN using universal phone modeling as the extra learning task (for a total of 4 learning tasks);
- *ML-MTL-DNN-UGS*: multi-lingual graphemic MTL-DNN using universal grapheme modeling as the extra learning task (for a total of 4 learning tasks); and
- *ML-MTL-DNN-UPS-UGS*: multi-lingual MTL-DNN using universal phone modeling and universal grapheme modeling as the extra learning tasks (for a total of 8 learning tasks).

All these DNNs were initialized from the same DBN which was pre-trained by training data from all the three languages. Afterwards, multiple softmax output layers were added to the DNNs, one for each learning task. Thus, the number of softmax layers in the various DNNs are: 1 for ML-STL-DNN, 3 for SHL-MDNN, 4 for ML-MTL-DNN-UPS/-UGS, and 8 for ML-MTL-DNN-UPS-UGS. As a result, during back-propagation fine-tuning, each training frame will activate 1, 1, 2, and 4 output nodes in ML-STL-DNN, SHL-MDNN, ML-MTL-DNN-UPS/-UGS, and ML-MTL-DNN-UPS-UGS respectively. Because of the use of multiple languages and MTL, some parts of the training procedure were modified. Firstly, the use of data from multiple languages requires the training utterances to be shuffled randomly so that the fine-tuning process would not be biased to a particular language at any time during training. Secondly, since more than one output node may be activated, the learning rate of the weights in the hidden layers were reduced by a factor equal to the number of activated output nodes. Otherwise, the training procedure is the same as that of MTL-DNN in Evaluation 1.

2) *Results and Discussions*: Table IX and Table X summarize the recognition performance of the various systems trained on the full training sets and the reduced training sets of all the three languages respectively. Performance of the previous mono-lingual STL-DNNs are repeated in the tables for comparison.

- The performance of multi-lingual STL-DNN (ML-STL-DNN) of the universal phones (graphemes) is far inferior to the triphone (trigrapheme) STL-DNN baseline. Similar finding was reported in [30]. Although the UPS/UGS models may share data among the various languages, the data become impure and they may fail to model the language specificities. Moreover, co-articulatory effects were not modeled as the targets in our ML-STL-DNNs are only monophones/monographemes states.
- On the other hand, multi-lingual models based on SHL-MDNN outperform their STL-DNN counterparts and reduce the WER by ~ 2 –11% relative in the full training set and ~ 4 –10% relative in the reduced training set. The improvements agree fairly well with the findings in [22] where the WER reductions are ~ 3 –5% relative. It is believed that the shared internal representation captures cross-lingual knowledge among the training languages.
- The multi-lingual MTL-DNN (ML-MTL-DNN) with an extra UPS (UGS) output layer further outperforms the corresponding phonetic (graphemic) SHL-MDNN. For example, in the case of reduced training set, the WER re-

duction improves from ~ 4 –10% relative in SHL-MDNN to ~ 7 –12% relative in ML-MTL-DNN-UPS/-UGS. In SHL-MDNN, the benefit of MTL is achieved only by learning common weights in the hidden layers, whereas in our ML-MTL-DNN-UPS/-UGS, the learning of the weights in the output layer of each language is further regularized by the learning of the weights in the output layer of UPS (UGS).

- Finally, based on the results of Evaluation 1 and above, we put the learning of phoneme models and grapheme models of the three languages together with the UPG and UGS, and obtained the best results that reduce the WER by ~ 6 –22% relative in the full set and ~ 8 –13% in the reduced set over the STL-DNN baselines. The improvements obtained by the ML-MTL-DNN-UGS-UPS are about twice of that from the respective SHL-MDNNs. All these are obtained without additional language resources.

TABLE IX
LWAZI: WERS (%) OF **MULTI-LINGUAL** SYSTEMS TRAINED ON THE FULL TRAINING SETS. FIGURES IN [] ARE WER REDUCTIONS (%) OVER THE MONO-LINGUAL PHONETIC STL-DNN-HMM BASELINE.

State Model	Units	Afrikaans	Sesotho	siSwati
mono-lingual STL-DNN	triphones	7.2	20.1	18.0
	trigraphemes	8.0	20.4	18.2
ML-STL-DNN	UPS phones	13.1	27.6	19.4
	UGS graphemes	19.4	27.9	22.2
SHL-MDNN	triphones	6.4 [11.1]	19.7 [2.00]	17.8
	trigraphemes	7.8	19.9	17.6 [2.22]
ML-MTL-DNN				
-UPS -UGS	triphones only	6.1 [15.3]	19.0 [5.47]	17.3
	trigraphemes only	7.5	19.1	17.0 [5.56]
-UPS-UGS	triphones only	5.6 [22.2]	18.9 [5.97]	17.2
	trigraphemes only	7.3	19.0	16.8 [6.67]

TABLE X
LWAZI: WERS (%) OF **MULTI-LINGUAL** SYSTEMS TRAINED ON ~ 1 -HOUR SMALL TRAINING SETS. FIGURES IN [] ARE WER REDUCTIONS (%) OVER THE MONO-LINGUAL PHONETIC STL-DNN-HMM BASELINE.

State Model	Units	Afrikaans	Sesotho	siSwati
mono-lingual STL-DNN	triphones	9.5	23.1	21.4
	trigraphemes	11.8	23.5	19.8
ML-STL-DNN	UPS phones	14.5	29.5	20.7 [3.27]
	UGS graphemes	25.0	29.6	22.6
SHL-MDNN	triphones	9.1 [4.21]	22.0 [4.76]	20.2
	trigraphemes	11.4	23.1	19.3 [9.81]
ML-MTL-DNN				
-UPS -UGS	triphones only	8.6 [9.47]	21.5 [6.93]	19.3
	trigraphemes only	11.2	22.2	18.8 [12.1]
-UPS-UGS	triphones only	8.3 [12.6]	21.3 [7.79]	19.0
	trigraphemes only	11.0	22.0	18.7 [12.6]

VI. CONCLUSIONS

Lack of data and language resources is the largest obstacle in low-resource ASR. In this paper, we propose two methods in the multi-task learning (MTL) framework using deep neural networks (DNNs) to train phonetic models in low-resource

languages *without* requiring additional resources. The resulting phonetic models are believed to generalize better to unseen data because the extra learning task(s) can exploit extra information from the training data to provide a representation bias to the original phonetic modeling task. This is made possible because both the inputs and hidden layers are shared by the multiple learning tasks. More specifically, for single-language low-resource ASR, we propose using grapheme modeling as the additional learning task to learn the language's phone models using an MTL-DNN. The proposed method was shown to work well not only for three low-resource South African languages, but also equally well for TIMIT phoneme recognition even though it is well-known that the grapheme-to-phoneme mappings in English are not simple. Moreover, although the method is originally designed for low-resource ASR, we further show that it works even for the WSJ large-vocabulary ASR task where there are adequate amount of training data. Thus, we believe the method can be applied in other general ASR tasks. Secondly, when the phone models of multiple low-resource languages are trained together, we propose using the acoustic modeling of a set of universal phones/graphemes (UPS/UGS) as the additional learning task. From the optimization perspective, the UPS task serves as a regularizer for the phonetic modeling of all the involved languages. From the language perspective, it forces the multi-lingual MTL-DNN to implicitly encode a mapping among the phonemes of all the languages. Finally, by combining the two methods, we are able to reduce the WERs of mono-lingual STL-DNN baselines by ~ 8 – 13% relative when only an hour of training data is available from each of the three South African languages, and ~ 7 – 22% relative when 3–8 hours of data are available. Additional memory and computational requirements are only required during MTL training; during recognition, the softmax layer(s) due to any extra tasks may be discarded. Furthermore, since our multi-lingual MTL-DNN has the same architecture as the multi-lingual SHL-MDNN but performs better than the latter, and the latter had been shown to be effective in cross-lingual model adaptation [21], [22], [42], we believe that our multi-lingual MTL-DNN will also perform better in cross-lingual model adaptation as well.

Our proposed MTL methods aim at improving the generalization of phonetic DNNs. There are many other ways to do this, and perhaps the most well-known one is the dropout method [52] which had been applied successfully to low-resource ASR [53]. Both our MTL methods and dropout are regularization methods but they use different mechanisms: dropout prevents overfitting by efficiently and approximately combining an exponentially many different neural network architectures, whereas our MTL methods exploit extra information from the data using additional learning task(s) which share(s) some commonality with the primary learning task and provide(s) a representation bias towards a better local optimum. Other ways such as weight pruning [54] and large-margin optimization [55], [56] have also been proposed, and it will be interesting to see if these methods are complementary to our proposed MTL methods.

Multi-task learning can be a powerful learning method if the tasks involved are truly related. In this paper, the multiple

tasks are carefully sought and their positive relationships are assumed, based on common knowledge. In the future, we would like to formulate the task relationships mathematically and make use of them in the MTL algorithm to further improve the ensuing model. In the machine learning community, this is known as multi-task relationship learning (MTRL), and MTRL for simple linear regression tasks had been investigated [57], [58]. How to do MTRL for complex tasks like speech recognition using DNN needs further investigation.

REFERENCES

- [1] S. Hunnicutt, H. M. Meng, S. Seneff, and V. W. Zue, "Reversible letter-to-sound sound-to-letter generation based on parsing word morphology," in *Proceedings of Eurospeech*, 1993, pp. 763–766.
- [2] E. G. Schukat-Talamazzini, H. Niemann, W. Eckert, T. Kuhn, and S. Rieck, "Automatic speech recognition without phonemes," in *Proceedings of Eurospeech*, 1993.
- [3] S. Kanthak and H. Ney, "Context-dependent acoustic modeling using graphemes for large vocabulary speech recognition," in *Proceedings of ICASSP*, vol. 1, 2002, pp. 845–848.
- [4] P. Charoenpornsawat, S. Hewavitharana, and T. Schultz, "Thai grapheme-based speech recognition," in *Proceedings of HLT-NAACL, Companion Volume: Short Papers*. ACL, 2006, pp. 17–20.
- [5] S. Stüker, "Acoustic modeling for under-resourced languages," Ph.D. dissertation, University of Karlsruhe, Germany, 2009.
- [6] T. Ko and B. Mak, "Eigentrigraphemes for under-resourced languages," *Speech Communication*, vol. 56, pp. 132–141, 2014.
- [7] S. Takahashi and S. Sagayama, "Four-level tied-structure for efficient representation of acoustic modeling," in *Proceedings of ICASSP*, vol. 1, 1995, pp. 520–523.
- [8] K. F. Lee, "Context-dependent phonetic hidden Markov models for speaker-independent continuous speech recognition," *IEEE Transactions on Speech and Audio Processing*, vol. 38, pp. 599–609, 1990.
- [9] S. J. Young and P. C. Woodland, "The use of state tying in continuous speech recognition," in *Proceedings of Eurospeech*, vol. 3, 1993, pp. 2203–2206.
- [10] M. Y. Hwang and X. D. Huang, "Shared-distribution hidden Markov model for speech recognition," *IEEE Transactions on Speech and Audio Processing*, vol. 1, pp. 414–420, 1993.
- [11] E. Bocchieri and B. Mak, "Subspace distribution clustering hidden Markov model," *IEEE Transactions on Speech and Audio Processing*, vol. 9, pp. 264–275, 2001.
- [12] J. R. Bellegarda and D. Nahamoo, "Tied mixture continuous parameter modeling for speech recognition," *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 38, no. 12, pp. 2033–2045, December 1990.
- [13] X. Huang and M. A. Jack, "Semi-continuous hidden Markov models for speech signals," *Computer Speech and Language*, vol. 3, no. 3, pp. 239–251, July 1989.
- [14] D. Povey, L. Burget, M. Agarwal, P. Akyazi, K. Feng, A. Ghoshal, O. Glembek, N. K. Goel, M. Karafiát, A. Rastrow *et al.*, "Subspace Gaussian mixture models for speech recognition," in *Proceedings of ICASSP*, 2010, pp. 4330–4333.
- [15] G. Saon and J.-T. Chien, "Bayesian sensing hidden Markov models," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 20, pp. 43–54, Jan. 2012.
- [16] M. J. F. Gales and K. Yu, "Canonical state models for automatic speech recognition," in *Proceedings of Interspeech*, 2010, pp. 58–61.
- [17] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Transactions on Knowledge and Data Engineering*, vol. 22, no. 10, pp. 1345–1359, 2010.
- [18] W. Byrne, P. Beyerlein, J. M. Huerta, S. Khudanpur, B. Marthi, J. Morgan, N. Peterek, J. Picone, D. Vergyri, and T. Wang, "Towards language independent acoustic modeling," in *Proceedings of ICASSP*, vol. 2, 2000, pp. 1029–1032.
- [19] V. Le and L. Besacier, "Automatic speech recognition for under-resourced languages: Application to Vietnamese language," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 17, pp. 1471–1482, 2009.
- [20] J. Kohler, "Multi-lingual phoneme recognition exploiting acoustic-phonetic similarities of sounds," in *Proceedings of ICSLP*, 1996.
- [21] A. Ghoshal, P. Swietojanski, and S. Renals, "Multilingual training of deep-neural networks," in *Proceedings of ICASSP*, 2013, pp. 7319–7323.

- [22] J.-T. Huang, J. Li, D. Yu, L. Deng, and Y. Gong, "Cross-language knowledge transfer using multilingual deep neural network with shared hidden layers," in *Proceedings of ICASSP*, 2013, pp. 7304–7308.
- [23] J. Kohler, "Language adaptation of multilingual phone models for vocabulary independent speech recognition tasks," in *Proceedings of ICASSP*, vol. 1, May 1998, pp. 417–420.
- [24] F. Grezl, M. Karafiat, and K. Vesely, "Adaptation of multilingual stacked bottle-neck neural network structure for new language," in *Proceedings of ICASSP*, May 2014, pp. 7654–7658.
- [25] D. Imseng, H. Bourlard, J. Dines, P. Garner, and M. Magimai-Doss, "Applying multi- and cross-lingual stochastic phone space transformations to non-native speech recognition," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 21, no. 8, pp. 1713–1726, Aug 2013.
- [26] N. T. Vu, D. Imseng, D. Povey, P. Motlicek, T. Schultz, and H. Bourlard, "Multilingual deep neural network based acoustic modeling for rapid language adaptation," in *Proceedings of ICASSP*, May 2014, pp. 7639–7643.
- [27] X. Cui, V. Goel, and B. Kingsbury, "Data augmentation for deep neural network acoustic modeling," in *Proceedings of ICASSP*, May 2014, pp. 5582–5586.
- [28] L. Burget, P. Schwarz, M. Agarwal, P. Akyazi, K. Feng, A. Ghoshal, O. Glembek, N. Goel, M. Karafiat, D. Povey, A. Rastrow, R. Rose, and S. Thomas, "Multilingual acoustic modeling for speech recognition based on subspace Gaussian mixture models," in *Proceedings of ICASSP*, March 2010, pp. 4334–4337.
- [29] P. Cohen, S. Dharanipragada, J. Gros, M. Monkowski, C. Neti, S. Roukos, and T. Ward, "Towards a universal speech recognizer for multiple languages," in *Proceedings of IEEE ASRU*, Dec 1997, pp. 591–598.
- [30] H. Lin, L. Deng, D. Yu, Y.-F. Gong, A. Acero, and C.-H. Lee, "A study on multilingual acoustic modeling for large vocabulary ASR," in *Proceedings of ICASSP*, April 2009, pp. 4333–4336.
- [31] R. Caruana, "Multitask learning," Ph.D. dissertation, Carnegie Mellon University, USA, 1997.
- [32] A. Mohamed, G. Dahl, and G. E. Hinton, "Acoustic modeling using deep belief networks," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 20, no. 1, pp. 14–22, 2012.
- [33] D. Chen, B. Mak, C.-C. Leung, and S. Sivasdas, "Joint acoustic modeling of triphones and trigraphemes by multi-task learning deep neural networks for low-resource speech recognition," in *Proceedings of ICASSP*, 2014.
- [34] S. Thrun and L. Pratt, *Learning to Learn*. Kluwer Academic Publishers, November 1997.
- [35] J. Baxter, "A model of inductive bias learning," *Journal of Artificial Intelligence Research*, vol. 12, pp. 149–198, 2000.
- [36] S. Ben-David and R. Schuller, "Exploiting task relatedness for multiple task learning," in *Proceedings of COLT*, 2003, pp. 567–580.
- [37] R. Collobert and J. Weston, "A unified architecture for natural language processing: Deep neural networks with multitask learning," in *Proceedings of ICML*. ACM, 2008, pp. 160–167.
- [38] G. Tur, "Multitask learning for spoken language understanding," in *Proceedings of ICASSP*, 2006, pp. 585–588.
- [39] Y. Huang, W. Wang, L. Wang, and T. Tan, "Multi-task deep neural network for multi-label learning," in *Proceedings of ICIP*, 2013, pp. 2897–2900.
- [40] S. Parveen and P. D. Green, "Multitask learning in connectionist ASR using recurrent neural networks," in *Proceedings of Eurospeech*, 2003, pp. 1813–1816.
- [41] M. Seltzer and J. Droppo, "Multi-task learning in deep neural networks for improved phoneme recognition," in *Proceedings of ICASSP*, 2013, pp. 6965–6968.
- [42] 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 ICASSP*, May 2013, pp. 8619–8623.
- [43] J. G. Fiscus, "A post-processing system to yield reduced word error rates: Recognizer output voting error reduction (ROVER)," in *Proceedings of IEEE ASRU*, 1997, pp. 347–354.
- [44] I. P. Association, *Handbook of the International Phonetic Association: A guide to the use of the International Phonetic Alphabet*. Cambridge University Press, 1999.
- [45] Meraka-Institute, *Lwazi ASR corpus*. <http://www.meraka.org.za/lwazi>, 2009.
- [46] "Lwazi phone set," <ftp://hlt.mirror.ac.za/Phoneset/Lwazi/Phoneset.1.2.pdf>, 2009.
- [47] T. Evgeniou and M. Pontil, "Regularized multi-task learning," in *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 2004, pp. 109–117.
- [48] M. M. Tempest, "Dictionarymaker 2.16 user manual," <http://dictionarymaker.sourceforge.net/>, 2009.
- [49] G. E. Hinton, S. Osindero, and Y. Teh, "A fast learning algorithm for deep belief nets," *Neural computation*, vol. 18, no. 7, pp. 1527–1554, 2006.
- [50] G. E. Dahl, D. Yu, L. 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, 2012.
- [51] V. Zue, S. Seneff, and J. Glass, "Speech database development at MIT: TIMIT and beyond," *Speech Communication*, vol. 9, no. 4, pp. 351–356, August 1990.
- [52] N. Srivastava, G. E. Hinton, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *Journal of Machine Learning Research*, 2014.
- [53] Y. Miao and F. Metze, "Improving low-resource CD-DNN-HMM using dropout and multilingual DNN training," in *Proceedings of ICASSP*, 2013.
- [54] D. Yu, F. Seide, G. Li, and L. Deng, "Exploiting sparseness in deep neural networks for large vocabulary speech recognition," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 2012.
- [55] Y. Tang, "Deep learning using linear support vector machines," 2013, arXiv:1306.0239v3 [cs.LG].
- [56] R. Min, Z. Yuan, D. A. Stanley, A. Bonner, and Z. Zhang, "A deep non-linear feature mapping for large-margin kNN classification," in *In Proceedings of ICDM*, 2009, pp. 357–366.
- [57] Y. Zhang and D.-Y. Yeung, "A convex formulation for learning task relationships in multi-task learning," *Proceedings of the 26th Conference on UAI*, July 2010.
- [58] W. Zhong and J. Kwok, "Convex multitask learning with flexible task clusters," in *Proceedings of ICML*, 2012.