JOINT ACOUSTIC MODELING OF TRIPHONES AND TRIGRAPHEMES BY MULTI-TASK LEARNING DEEP NEURAL NETWORKS FOR LOW-RESOURCE SPEECH RECOGNITION

Dongpeng Chen, Brian Mak

Department of Computer Science & Engineering
Hong Kong University of Science & Technology
{dpchen,mak}@cse.ust.hk

Cheung-Chi Leung, Sunil Sivadas

Institute for Infocomm Research
A*STAR, Singapore
{ccleung,sivadass}@i2r.a-star.edu.sg

ABSTRACT

It is well-known in machine learning that multitask learning (MTL) can help improve the generalization performance of singly learning tasks if the tasks being trained in parallel are *related*, especially when the amount of training data is relatively small. In this paper, we investigate the estimation of triphone acoustic models in parallel with the estimation of trigrapheme acoustic models under the MTL framework using deep neural network (DNN). As triphone modeling and trigrapheme modeling are highly related learning tasks, a better shared internal representation (the hidden layers) can be learned to improve their generalization performance. Experimental evaluation on three low-resource South African languages shows that triphone DNNs trained by the MTL approach perform significantly better than triphone DNNs that are trained by the single-task learning (STL) approach by ∼3-13%. The MTL-DNN triphone models also outperform the ROVER result that combines a triphone STL-DNN and a trigrapheme STL-DNN.

Index Terms: triphone modeling, trigrapheme modeling, multitask learning, deep neural networks

1. INTRODUCTION

Since the emergence of automatic speech recognition (ASR) techniques several decades ago, huge research efforts have been spent on the most popular languages such as English, French, German, Mandarin, ..., etc., and great achievement has been accomplished. On the other hand, there are still many languages in the world which do not benefit from the advanced human language technologies due to their lack of audio and language resources that are costly to obtain. Ways are sought to either create resources for a new language more efficiently, or to mitigate its reliance on language-specific resources in training its acoustic models. Notable efforts include cross-lingual [1, 2] and multilingual [3] acoustic modeling techniques. A basic assumption behind these methods is that a good mapping between phonemes in some rich-resource languages and the phonemes of the target low-resource language can be found so that transfer learning may be applied to transform the acoustic models of the former to those of the latter.

In this paper, we take a different approach: we try to improve the phonetic models of a low-resource language by using only its own language resources without relying on finding a good mapping between its phonemes and phonemes from other languages which is sometimes not easily achieved. We investigate a multitask learning (MTL) approach [4] in which the estimation of triphone acoustic models is performed in parallel with the estimation of trigrapheme acoustic models of the same language using deep neural network (DNN) [5]. According to the theory of multitask learning, *related tasks* can be jointly learned to improve the generalization performance of both tasks; the effect is more prominent when the amount of training data is relatively small. Obviously, triphone modeling and trigrapheme modeling are highly related learning tasks. Their joint training does *not* require additional resources of other languages but only the orthographic transcriptions of the training data as well as a phonetic dictionary of the target language which phonetic acoustic modeling already requires. In fact, both can be trained using the same kind of acoustic feature vectors (e.g., PLP coefficients in our case). Although the performance of grapheme-based acoustic modeling [6, 7, 8] in ASR can be sensitive to the language under investigation, it has been shown to be comparable to phone-based modeling on many languages too. Thus, there are reasons to believe that trigrapheme modeling may be used as the secondary task to improve the performance of triphone modeling — the primary task — in the MTL framework.

The rest of this paper is organized as follows. In the next section, the concepts of multitask learning and deep neural network are reviewed. Then in Section 3, we describe the proposed joint training of triphone and trigrapheme models using a DNN in the MTL framework. Experimental evaluation on three low-resource South African languages are described in Section 4, which is followed by concluding remarks in Section 5.

2. REVIEW OF MULTITASK LEARNING (MTL) AND DEEP NEURAL NETWORK (DNN)

2.1. Deep Neural Network

Deep neural network (DNN) is simply a multilayer perceptron with many hidden layers. Although the concept is not new, there was a resurgence of DNNs after Hinton et al. introduced a fast pre-training algorithm for a deep belief network (DBN) [9]. Since then DNN has been proved to be very effective in many tasks of speech recognition [5], computer vision [10] and natural language processing [11].

Theoretically, DNN is able to model highly non-linear functions but it is very hard to train DNNs in practice. Hinton proposed initializing a DNN with a generative pre-trained DBN, which consists of repeated layers of restricted Boltzmann machine (RBM). Each RBM is an undirected bipartite graph consisting of two disjoint groups of nodes: visible nodes and hidden nodes. RBM can be effectively trained by minimizing the contrastive divergence [9] in an unsupervised manner. After an RBM is trained, another new one is placed on top of it. At the end, several RBMs are stacked together to form a DBN which is then converted to a DNN with the addition of an output layer that is designed for each application. In ASR, the output layer is usually a softmax layer consisting of units that represent phones or pho-

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

netic states. Finally supervised backpropagation is performed on the whole network to optimize the per-frame cross-entropy between the predictions and the targets.

2.2. Multitask Learning

Multitask learning (MTL) [4] or learning to learn [12] 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 can 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 generalizes better for future unseen data, and the amount of training data is effectively increased for each task. In [13], a statistical learning theory based approach to MTL is developed and a generalization bound on the average error of MTL is derived. In [13, 14], the notion of relatedness among multiple tasks is defined in a particular way so as to derive a tighter generalization bound for each learning task. In his thesis [4], Caruana postulates some requirements for related tasks if their joint learning in the MTL approach is 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-backpropagation.

MTL has been applied successfully in many speech, language, and image/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, in ASR, MTL is used to improve ASR robustness using recurrent neural networks in [15]. In language applications, [11] applies MTL on a single convolutional neural network to produce state-of-the-art performance for several language processing predictions; [16] improves intent classification in goal-oriented human-machine spoken dialog systems especially when the amount of labeled training data is limited. In [17], the MTL approach is used to perform multi-label learning in an image annotation application.

2.3. Multitask Learning Deep Neural Network (MTL-DNN)

Obviously one may apply MTL with the recently very successful DNN to further improve learning performance. Related works in the area of ASR include the use of MTL-DNN for TIMIT phoneme recognition in [18] which learns posteriors of monophone states together with a secondary task which can be learning phone labeling, state context, or phone context. MTL-DNN is also used in multilingual ASR to transfer cross-lingual knowledge [19, 20]. In these works, during pre-training and subsequent fine-tuning, the hidden layers are updated with data from multiple languages, but each language has its own softmax layer that estimates the posterior probabilities of its senones (tied-states).

3. JOINT TRIPHONE AND TRIGRAPHEME ACOUSTIC MODELING WITH MTL-DNN

We would like to improve the generalization performance of triphone models by jointly training them with the trigrapheme models of the same language under the MTL framework with a DNN for low-resource languages. The motivations are:

- triphone modeling and trigrapheme modeling are obviously related learning tasks for the same language;
- when they are trained singly, they give comparable recognition performance;
- they can be trained using the same acoustic input features, and no additional language resources are required besides those already used by triphone modeling;

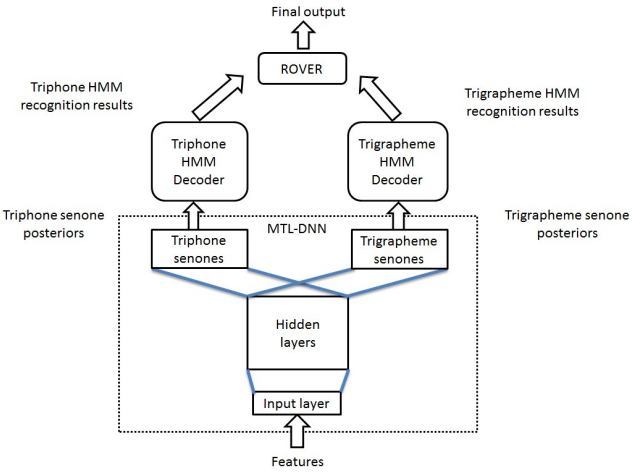

Fig. 1. An MTL-DNN system for joint training of triphone and trigrapheme acoustic models.

- when DNN is used for their joint training, they share the same internal representation (the hidden layers);
- MTL is particularly helpful when the amount of training data is limited.

3.1. The Basic MTL-DNN

Since in our experience, triphone models usually perform at least as well as trigrapheme models, we pick triphone acoustic modeling as the primary task, and trigrapheme acoustic modeling as the secondary task for MTL. Fig.1 shows an overview of the proposed MTL-DNN system for joint training of triphone and trigrapheme acoustic models. The DNN architecture is similar to the one used in multi-lingual speech recognition [19, 20]. 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 a given input acoustic frame. More specifically, given an input vector $\mathbf{x}$, the posterior probability of the ith triphone senone s_i at the triphone output layer is computed using the softmax function as follows:

$$P(s_i^{(p)}|\mathbf{x}) = \frac{\exp(y_i^{(p)})}{\sum_{i'=1}^{N^{(p)}} \exp(y_{i'}^{(p)})}, \quad \forall i = 1, \ldots, N^{(p)},$$

where $y_i^{(p)}$ is the activation of the senone, and $N^{(p)}$ is the total number of triphone senones. A similar formula may be derived for the posterior probabilities $P(s_i^{(g)}|\mathbf{x})$ of the $N^{(g)}$ trigrapheme senones. Finally, the whole MTL-DNN is trained by minimizing the sum of cross-entropies from each of the two tasks over all frames:

$$-\sum_{\mathbf{x}} \left(\sum_{i=1}^{N^{(p)}} d_i^{(p)} \log P(s_i^{(p)}|\mathbf{x}) + \sum_{i=1}^{N^{(g)}} d_i^{(g)} \log P(s_i^{(g)}|\mathbf{x}) \right),$$

where $d_i^{(p)}$ and $d_i^{(g)}$ are the target values of the ith triphone senone and the ith trigrapheme senone respectively.

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

aligning the training data. During MTL-DNN training, the target values of one triphone senone output unit and one trigrapheme senone output unit will be set to 1.0. During decoding, the MTL-DNN posterior probabilities of the senones are first converted back to scaled senone likelihoods by dividing them by the senone priors as follows:

$$P(\mathbf{x}|s_i^{(\tau)}) \propto \frac{P(s_i^{(\tau)}|\mathbf{x})}{P(s_i^{(\tau)})}, \quad \forall i = 1, \ldots, N^{(\tau)}, \text{ and } \tau = \{p, g\}.$$

Afterward, Viterbi decoding is performed on the respective MTL-DNN-HMM.

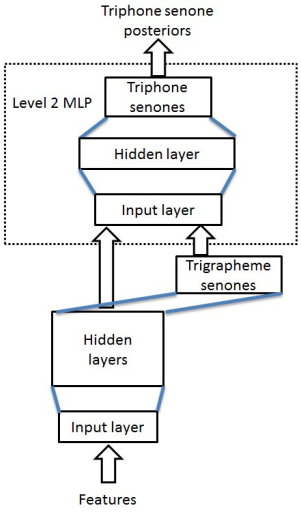

Fig. 2. MTL-DNN2: Stacking an STL-MLP on top of the MTL-DNN system of Fig. 1.

3.2. Extended MTL-DNN with an STL-MLP (Triphone MTL-DNN2)

We further investigate if the trigrapheme posteriors that are obtained as a by-product of MTL may be useful features for triphone modeling when the amount of training data is small. In a manner similar to the use of NN tandem features in HMM training [21], we concatenate the outputs from the shared hidden layers with the trigrapheme senone posteriors from the well-trained MTL-DNN and feed them to another STL multi-layer perceptron (MLP) to estimate the triphone posteriors again. The MLP has only a single hidden layer with 2048 units, and an output layer with triphone senone targets. Back-propagation is performed to train this MLP while keeping the MTL-DNN unchanged. The corresponding system, which we call *triphone MTL-DNN2*, is shown in Fig. 2.

3.3. System Combination by ROVER

We also compare the gain from transfer learning between tasks in MTL with the gain obtained from the simple system combination method ROVER [22]. If the decoding errors of the triphone-based and the trigrapheme-based systems are complementary, ROVER may be able to improve recognition performance by integrating their outputs. The SRILM toolkit [23] was employed to do posterior decoding on the two n-best lattices produced by the two systems while estimating the confidence scores of the decoded words. Finally, ROVER aligned multiple hypotheses from the two systems by dynamic programming, and searched for the best path based on the confidence scores of the words.

Table 1. Number of phones and graphemes of the 3 languages, and the test-set perplexity of their LMs.

Data Set	#Phones	#Graphemes	LM Perplexity
Afrikaans	37	31	11.18
Sesotho	41	25	19.69
siSwati	40	25	10.94

Table 2. Partition and details of various 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	0.00%
Train	160	4,784	3.37	1,513	0.00%
Dev	20	600	–	870	0.89%
Eval	20	599	–	876	0.97%
Sesotho:					
Train-S	162	1,206	1.43	1,513	0.00%
Train	162	4,826	5.70	2,360	0.00%
Dev	20	600	–	1,096	1.86%
Eval	20	601	–	1,089	2.29%
siSwati:					
Train-S	156	580	1.02	1,833	0.00%
Train	156	4,643	8.38	4,645	0.00%
Dev	20	599	–	1,889	6.14%
Eval	20	596	–	1,851	4.53%

4. EXPERIMENTAL EVALUATION

The proposed MTL-DNN approach is evaluated on three low-resource South African languages.

4.1. The Lwazi Speech Corpus

The Lwazi ASR corpus [24] consists of telephone speech for all the 11 official languages of South Africa. For each language, a 5000-word pronunciation dictionary was created. These dictionaries cover the most common words in the languages but not all the words in the corpus. Thus, for the phone-based experiments, the DictionaryMaker [25] software was used to generate dictionary entries for the words that are not covered by the Lwazi dictionaries.

Three languages were selected from the corpus in our experiments. They are Afrikaans, Sesotho, and siSwati. The numbers of phones and graphemes of the three languages, together with the test-set perplexity of their word bigram language models (which were trained only by the transcriptions in the training set) are shown in Table 1. The partition of the various data sets into training, development, and test subsets follows from [26]. In order to evaluate the proposed joint triphone and trigrapheme modeling in the scenario where acoustic data is scarce and good pronunciation dictionary may not be available, smaller data sets were created by randomly sampling approximately 1 hour of speech from the full training set of each language; care had been taken to ensure that each speaker has roughly the same number of utterances. Details of the various data sets are listed in Table 2.

4.2. Feature Extraction and System Configurations

An input acoustic vector consists of the first 13 PLP coefficients, including c0, and their first and second order derivatives. These 39-dimensional feature vectors were extracted at every 10ms over a window of 25ms. Speaker-based cepstral mean subtraction

and variance normalization were performed. Then, conventional strictly left-to-right 3-state continuous-density hidden Markov models were trained by maximum-likelihood estimation. State output probability density functions were modeled by Gaussian mixture densities with at most 16 components.

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 had 4 hidden layers with 2048 nodes per layer. During pre-training, the mini-batch size was kept at 128, 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. 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 increased.

Each MTL-DNN was initialized by the DBN of the corresponding STL-DNN. But now the single softmax output layer in STL-DNN was replaced by 2 separate softmax layers, one for the primary task and one for the secondary task. Otherwise, the training procedure was the same as that of STL-DNN. Then, the MTL-DNN was extended by stacking on top of it an STL-MLP layer which takes the trigrapheme posterior probabilities and the outputs from the last hidden layer of the well-trained MTL-DNN to train the final triphone senone posteriors. This experiment was only performed with the small reduced training data sets.

4.3. Experimental Results

Experimental results on both the reduced and full training data sets of each language are listed in Table 3 and 4 respectively. We have the following observations:

- For GMM-HMMs, trigrapheme models are superior to triphone models in siSwati and Sesotho when there are only about 1 hour of training data. One reason may be that there are much fewer grapheme units than phoneme units in the two languages: the ratio is 1:1.6 in these two languages but is 1:1.2 in Africaans. Thus, the trigrapheme models could be robustly trained using a smaller amount of data. This is supported by the fact that the better performance disappear when the full training set was used.
- All phone-based and grapheme-based DNN-HMMs outperform their GMM-HMM counterparts by 15–24% in the reduced training sets and 9–24% in the full training sets.
- Triphone and trigrapheme models estimated jointly by MTL-DNN consistently outperform their respective STL-DNN counterparts: the gain ranges from 2% to 13%. This shows that MTL benefits learning of not only the primary task but also the secondary task.
- The triphone MTL-DNNs even outperform the ROVER integration of triphone and trigrapheme STL-DNNs (except in one case when the two are basically the same). This shows that MTL can transfer knowledge between the multiple learning tasks to improve recognition performance, and such knowledge sharing is more effective than ROVER integration. Nevertheless, ROVER may still take advantage of residual complementary errors made by the triphone and trigrapheme MTL-DNN-HMMs and gives the best recognition performance by integrating them.

Table 3. Word recognition accuracies (%) with the small training sets (∼1 hr). Figures in bracket are #tied states in each baseline.

Model	Afrikaans	Sesotho	siSwati
triphone GMM	87.5 (514)	70.0 (722)	72.9 (271)
trigrapheme GMM	85.5 (210)	72.3 (324)	75.4 (243)
triphone STL-DNN	90.5	76.9	78.6
trigrapheme STL-DNN	88.2	76.5	80.2
triphone MTL-DNN	**91.1**	**77.9**	79.4
trigrapheme MTL-DNN	88.7	76.9	**81.1**
triphone MTL-DNN2	**91.3**	**78.1**	**81.2**
ROVER on STL-DNNs	90.8	77.6	80.7
ROVER on MTL-DNNs	**91.3**	**78.2**	**81.6**

Table 4. Word recognition accuracies (%) with the full training sets. Figures in bracket are #tied states in each baseline.

Model	Afrikaans	Sesotho	siSwati
triphone GMM	90.7 (641)	75.6 (741)	79.8 (339)
trigrapheme GMM	89.4 (728)	75.7 (543)	80.0 (931)
triphone STL-DNN	92.8	79.9	82.0
trigrapheme STL-DNN	92.0	79.6	81.8
triphone MTL-DNN	**93.6**	**80.5**	**82.5**
trigrapheme MTL-DNN	92.4	80.2	82.0
ROVER on STL-DNNs	93.3	80.3	82.6
ROVER on MTL-DNNs	**93.8**	**80.7**	**83.0**

- Triphone MTL-DNN2 trained with the reduced small training set gives a performance that is almost as good as the ROVER integration of the triphone and trigrapheme MTL-DNNs, even on siSwati where triphone DNNs are inferior to the corresponding trigrapheme DNNs.

5. CONCLUSIONS AND RELATION TO PRIOR WORK

We make use of the fact that triphone modeling and trigrapheme modeling are highly related learning tasks, and successfully apply multitask learning (MTL) to joint training of the two models using deep neural networks (DNN). The ensuing triphone MTL-DNN outperforms not only its STL-DNN counterpart, but also the ROVER system that integrates the triphone and trigrapheme STL-DNNs. Our work is similar to the use of MTL on multilingual ASR [19, 20] but we do not require any additional language resources other than those for training the primary task (triphone modeling here). Moreover, each training input vector in [19, 20] will only activate one target of one language and is not shared by other languages, whereas each input vector in our work actually activates one senone in the output layer of each of the multiple tasks. From this perspective, our work is similar to the use MTL for TIMIT phoneme recognition in [18]. However, we use grapheme information instead of other phonetic information in [18], and we focus on word recognition instead.

We further investigate stacking up another simple STL-MLP on top of the well-trained MTL-DNN and obtain triphone models that perform close to the ROVER system that integrates the triphone and trigrapheme MTL-DNNs. This opens the possibility of further improving the MTL-DNN by the addition of other structures such as additional layers of MTL-DNNs.

6. ACKNOWLEDGMENTS

This work was supported by the Research Grants Council of the Hong Kong SAR under the grant number HKUST616513.

7. REFERENCES

[1] K. U. Ogbureke and J. Carson-Berndsen, "Framework for cross-language automatic phonetic segmentation," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 2010, pp. 5266–5269.

[2] 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, no. 8, pp. 1471–1482, 2009.

[3] J. Kohler, "Multi-lingual phoneme recognition exploiting acoustic-phonetic similarities of sounds," in *Proceedings of the International Conference on Spoken Language Processing*, 1996, pp. 1029–1032.

[4] R. Caruana, *Multitask Learning*, Ph.D. thesis, Carnegie Mellon University, USA, 1997.

[5] A. Mohamed, G.E. 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.

[6] S. Kanthak and H. Ney, "Context-dependent acoustic modeling using graphemes for large vocabulary speech recognition," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*. IEEE, 2002, vol. 1, pp. 845–848.

[7] P. Charoenpornsawat, S. Hewavitharana, and T. Schultz, "Thai grapheme-based speech recognition," in *Proceedings of the Human Language Technology Conference of the NAACL, Companion Volume: Short Papers*. Association for Computational Linguistics, 2006, pp. 17–20.

[8] S. Stüker, *Acoustic Modeling for Under-Resourced Languages*, Ph.D. thesis, University of Karlsruhe, Germany, 2009.

[9] 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.

[10] H. Lee, R. Grosse, R. Ranganath, and A. Y. Ng, "Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations," in *Proceedings of the International Conference on Machine Learning*. ACM, 2009, pp. 609–616.

[11] R. Collobert and J. Weston, "A unified architecture for natural language processing: Deep neural networks with multitask learning," in *Proceedings of the International Conference on Machine Learning*. ACM, 2008, pp. 160–167.

[12] S. Thrun and L. Pratt, *Learning to Learn*, Kluwer Academic Publishers, November 1997.

[13] J.Baxter, "A model of inductive bias learning," *Journal of Artificial Intelligence Research*, vol. 12, pp. 149–198, 2000.

[14] S. Ben-David and R. Schuller, "Exploiting task relatedness for multiple task learning," in *Proceedings of COLT*, 2003, pp. 567–580.

[15] S. Parveen and P. D. Green, "Multitask learning in connectionist ASR using recurrent neural networks," in *Proceedings of the European Conference on Speech Communication and Technology*, 2003, pp. 1813–1816.

[16] G. Tur, "Multitask learning for spoken language understanding," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 2006, pp. 585–588.

[17] Y. Huang, W. Wang, L. Wang, and T. Tan, "Multi-task deep neural network for multi-label learning," in *Proceedings of the IEEE International Conference on Image Processing*, 2013, pp. 2897–2900.

[18] M. Seltzer and J. Droppo, "Multi-task learning in deep neural networks for improved phoneme recognition," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 2013, pp. 6965–6968.

[19] Jui-Ting Huang, Jinyu Li, Dong Yu, Li Deng, and Yifan Gong, "Cross-language knowledge transfer using multilingual deep neural network with shared hidden layers," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 2013, pp. 7304 – 7308.

[20] A. Ghoshal, P. Swietojanski, and S. Renals, "Multilingual training of deep-neural netowrks," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 2013, pp. 7319–7323.

[21] H. Hermansky, D. P. W. Ellis, and S. Sharma, "Tandem connectionist feature extraction for conventional hmm systems," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*. IEEE, 2000, vol. 3, pp. 1635–1638.

[22] J. G. Fiscus, "A post-processing system to yield reduced word error rates: Recognizer output voting error reduction (ROVER)," in *Automatic Speech Recognition and Understanding, 1997. Proceedings., 1997 IEEE Workshop on*. IEEE, 1997, pp. 347–354.

[23] A. Stolcke et al., "SRILM–an extensible language modeling toolkit.," in *INTERSPEECH*, 2002, pp. 901–904.

[24] Meraka-Institute, "Lwazi ASR corpus," `http://www.meraka.org.za/lwazi`", 2009.

[25] M.Davel M. Tempest, "Dictionarymaker 2.16 user manual," `http://dictionarymaker.sourceforge.net/`, 2009.

[26] T. Ko and B. Mak, "Eigentrigraphemes for under-resourced languages," *Speech Communication*, vol. 56, pp. 132–141, 2014.