INTERSPEECH 2016
September 8–12, 2016, San Francisco, USA

Automatic Pronunciation Generation by Utilizing a Semi-supervised Deep Neural Networks

Naoya Takahashi[1], *Tofigh Naghibi*[2], *Beat Pfister*[2]

[1]Sony Corporation, Japan
[2]Speech Processing Group, ETH Zurich, Switzerland
NaoyaA.Takahashi@jp.sony.com, {naghibi, pfister}@tik.ee.ethz.ch

Abstract

Phonemic or phonetic sub-word units are the most commonly used atomic elements to represent speech signals in modern ASRs. However they are not the optimal choice due to several reasons such as: large amount of effort required to handcraft a pronunciation dictionary, pronunciation variations, human mistakes and under-resourced dialects and languages. Here, we propose a data-driven pronunciation estimation and acoustic modeling method which only takes the orthographic transcription to jointly estimate a set of sub-word units and a reliable dictionary. Experimental results show that the proposed method which is based on semi-supervised training of a deep neural network largely outperforms phoneme based continuous speech recognition on the TIMIT dataset.

Index Terms: speech recognition, deep neural networks, semi-supervised learning, dictionary, sub-word unit, k-dimensional Viterbi

1. Introduction

The three principal resources typically required for developing a phoneme based automatic speech recognizer (ASR) are: transcribed acoustic data for acoustic model estimation, text data for language model estimation, and a pronunciation dictionary to map words to sequences of sub-word units. Manual preparation of such resources requires significant investment and expertise. Therefore, an automatic generation of pronunciation dictionary from the data is clearly required for many dialects and languages.

Developing ASRs for dialects and under-resourced languages has attracted growing attention over the past few years [1, 2, 3]. A main challenge to develop ASR for under-resourced domains is to produce a reliable pronunciation dictionary from limited available resources. For major languages, however, a canonical pronunciation dictionary is usually already available. However, such dictionaries may be error-prone due to the fact that they are manually generated and in most cases do not cover pronunciation variants. There were several attempts to tackle these problems [4, 5, 6, 7].

Lu et al. [8] proposed a data-driven dictionary generator to include new pronunciations based on newly coming acoustic evidence. Goel et al. in [9] use a grapheme-to-phoneme approach to guess the pronunciation and iteratively refine the acoustic model and the dictionary. However, these methods still require a high-quality initial pronunciation dictionary created by an expert.

In modern ASRs words are represented by smaller sub-word units such as phonemes and the pronunciation dictionary maps words to sequences of sub-word units. However, sub-word units do not essentially need to be linguistically motivated elements. In fact, given a set of acoustic samples, the linguistically defined units are most probably not the optimal ones for speech recognition [10]. For instance telephony speech, where high frequency components have been filtered out, requires a modified dictionary with slightly different set of fricatives than full-bandwidth speech.

Over the past few years, there have been several attempts to move beyond phoneme based sub-word units by jointly learn a set of sub-word units and their corresponding dictionary directly from the given data [11, 12, 8]. Bacchiani and Ostendorf [12] proposed an iterative acoustic segmentation and clustering approach to build sub-word units from speech signals and subsequently construct the dictionary based on the estimated sub-word units. Singh et al. [8] introduced a divide-and-conquer strategy to recursively update sub-word units and dictionary. The dictionary computation was done by means of an n-best type algorithm which is known to produce sub-optimal solutions. Although their approach demonstrates some promising results, the performance is still not comparable with a phoneme based ASR.

The main focus of this paper is to design an ASR based on an automatically generated dictionary that outperforms commonly used phoneme based ASRs. While most of the solutions proposed to find a pronunciation based on multiple utterances of a word are n-best type heuristics [8, 13, 14], in this paper, we employ an approximation of the K-dimensional Viterbi algorithm proposed in our previous works [15, 10]. This approach gives us the maximum-likelihood estimates of the pronunciations. These high-quality pronunciations are one of the key factors to outperform phoneme based ASRs. Moreover, to learn proper sub-word units, we combine the strength of Gaussian mixture models (GMM) and deep neural network (DNN) based acoustic modeling. We formulate this problem as an instance of a semi-supervised self-learning process. By taking advantage of the robustness of hidden Markov models (HMM) with GMM based observation probability distribution against labeling errors, we train the first set of sub-word units and output the first set of pronunciations. We then use this dictionary to re-label the data and employ the higher expressiveness of DNNs to improve the modeling of sub-word units and the dictionary in an iterative process. In each iteration round, a new dictionary is generated and by means of this new dictionary the data is re-labeled. This data is again used to train the DNN. As shown in the experiments, the proposed results achieves more than 10% absolute improvement over the phoneme based approach on TIMIT data in a continuous speech recognition task.

The reminder of this paper is organized as follows. The proposed framework and its components for joint sub-word units

http://dx.doi.org/10.21437/Interspeech.2016-761

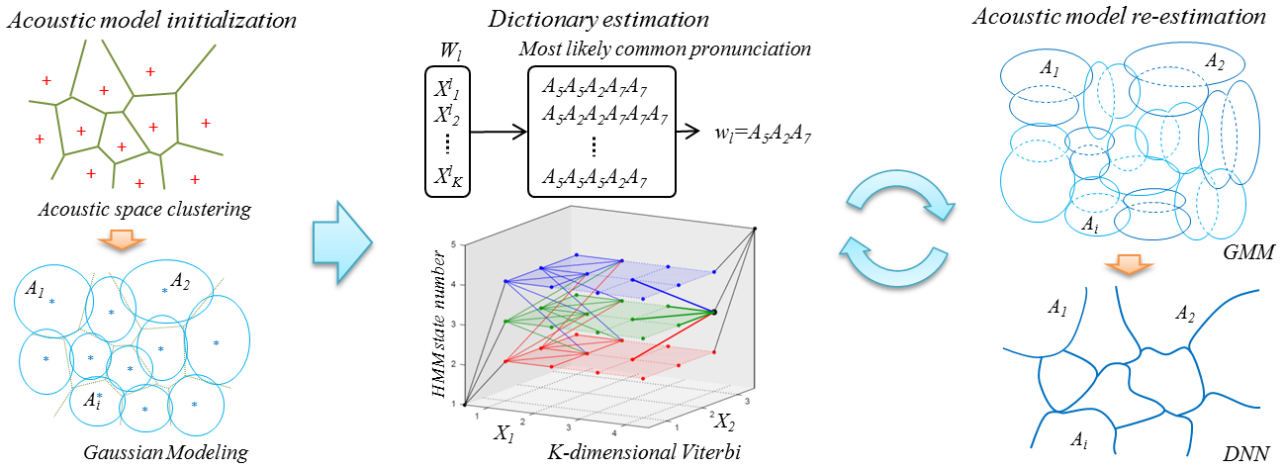

Figure 1: *Framework of joint sub-word and dictionary learning. K-dimensional Viterbi illustrated in case of* $K = 2$.

and dictionary learning are introduced in Section 2. In Section 3 the experimental results are demonstrated and finally, conclusions are summarized in Section 4.

2. Semi-supervised joint Dictionary and Acoustic Model Learning

2.1. Framework

In the rest of this paper, we refer to data-driven sub-word units as abstract acoustic elements (AAEs) in contrast to phones. Our goal is to jointly learn the pronunciation dictionary $d^* = \{\omega_1, \cdots, \omega_L\}$ of L pronunciations ω_i and N AAE models $\lambda^* = \{A_1, \cdots, A_N\}$ that maximize the joint likelihood:

$$\lambda^*, d^* = \arg\max_{\Lambda, D} P(\mathbf{X}|\mathbf{T}, \Lambda, D) \tag{1}$$

where $\mathbf{X} = (X_1, \cdots, X_M)$ is the set of training utterances, $\mathbf{T} = (T_1, \cdots, T_M)$ is the set of corresponding orthographic transcriptions, M is the number of utterances, Λ is the universe of all possible sets of N AAEs and D is the universe of all the dictionaries which map words to AAEs sequences. It is hard to find the optimal solution for the optimization problem in (1) due to its complex non-linear nature. It is thus decomposed into two simpler optimization problems which can be solved iteratively.

$$d^i = \arg\max_{D} P(X|T, \lambda^i, D) \tag{2}$$

$$\lambda^{i+1} = \arg\max_{\Lambda} P(X|T, \Lambda, d^i) \tag{3}$$

Since the pronunciation of each word can be estimated independently from other words, the dictionary estimation in (2) can be decomposed into L maximum likelihood estimations as follows:

$$\omega_l = \arg\max_{\omega} \prod_{j \in \Omega_l} \max_{\mathbf{S}_j} P(X_j, \mathbf{S}_j|\lambda) \quad \text{subject to: } \mathbf{S}_j \in \mathbb{S}_\omega \tag{4}$$

where Ω_l is the set of indices of utterances of word W_l, $\mathbf{S}_j$ is a sequence of AAEs and $\mathbb{S}_\omega$ denotes a set of all possible AAE sequences of the pronunciation ω. For instance in $\mathbb{S}_\omega$, if the pronunciation is $\omega = A_1A_2A_3$, some samples in $\mathbb{S}_\omega$ may be $A_1A_1A_1A_2A_3$, $A_1A_2A_2A_3A_3$ and $A_1A_1A_2A_3A_3A_3$. The constraint in (4) implies that all AAE sequences should be samples of the same pronunciation. For the case where λ is modeled by a left-to-right HMM without skips, which is the most common topology in HMM based ASRs, a solution of (4) has been proposed in [15] (Details are in Section 2.3.). In (3), since the dictionary is fixed, the problem results in a common acoustic model estimation given the dictionary. However, the labels reassigned by using the estimated dictionary are very noisy since the dictionary is automatically estimated from data without any expert supervision. Therefore, a robust model is required at early stage of the training iteration while a more expressive and powerful model such as a DNN [16, 17] can be used after the reliable dictionary is obtained.

The joint dictionary and AAE learning framework is illustrated in Figure 1 and summarized as follows:

Algorithm 1 Semi-supervised joint AAEs and dictionary learning

1: $i = 0$
// Initialize AAE models λ^0 (Section 2.2)
2: Clustering the acoustic space.
3: Model each cluster by GMM and set as λ^0.
// Start joint AAEs and dictionary learning
4: **while** (Performance is improved) **do**
5: Given AAE models λ^i, update dictionary d^i by maximizing joint likelihood multiple utterances (Section 2.3).
6: Given dictionary d^i, double the number of mixtures and update AAE models λ^{i+1} (Section 2.4).
7: $i \leftarrow i + 1$
8: **end while**
9: Replace GMM by DNN and train AAE model using labels obtained by HMM-GMM (Section 2.4).
10: **while** (Performance is improved) **do**
11: Given AAE models λ^i, update dictionary d^i by maximizing joint likelihood multiple utterances.
12: Given dictionary d^i, re-train DNN based AAE models λ^{i+1} (Section 2.4).
13: $i \leftarrow i + 1$
14: **end while**

2.2. Acoustic Model Initialization

Initial AAE models can simply be obtained by clustering the acoustic space. The acoustic space can be described by any feature as long as it is informative enough to discriminate between different words. We employed the Linde-Buzo-Gray (LBG) algorithm [18] with a squared-error distortion measure to cluster the acoustic feature vectors. The LBG clustering algorithm tends to assign more codebook vectors to high-density areas which is a useful property in order to obtain discriminative initial AAEs. Each cluster is then modeled by a GMM with a single Gaussian component. These models are used as the initial models for AAEs.

2.3. Dictionary Generation

The solution of (4) proposed in [15] is an extension of the standard one-dimensional Viterbi algorithm to K dimensions. The K-dimensional Viterbi algorithm calculates the most probable HMM state sequence which is common to K given utterances. While this algorithm is rigorous, its complexity grows exponentially with the number of utterances, which consequently makes it infeasible to apply it to more than a few utterances. An efficient approximation of the K-dimensional Viterbi algorithm has been proposed in [10] where the problem to find the joint alignment and the optimal common sequence for K utterances is decomposed into $K-1$ applications of two-dimensional Viterbi algorithm. This approximation starts with finding the best alignment between two utterances. Then, while keeping the alignment between the already processed utterances fixed, the next utterance is aligned with this master utterance. The AAE sequence of the final master utterance is the approximation of the K-dimensional Viterbi pronunciation.

2.4. Acoustic Modeling

Once the dictionary is updated, all utterances are decoded based on the new pronunciation of the words in the dictionary and the AAEs are re-estimated according to the new labels. The AAEs can be modeled by commonly used models such as HMM/GMM or HMM/DNN. However, at the beginning of the training iteration, the model and dictionary are not accurate enough and more probable to get stuck in a bad local optimum if the model's degree of freedom is too high. In order to avoid this situation, we start the training with a simple model, namely one Gaussian component for each AAE with a diagonal covariance matrix. In each iteration, the dictionary gets more accurate. Thus, the number of mixture components are doubled in order to increase the modeling power. Once the performance is saturated the GMM is replaced with the DNN in order to utilize more expressive modeling capability. This process makes the semi-supervised DNN training feasible and prevents it to get stuck in a bad local optimum. The HMM state-level transcription is obtained by force-aligned decoding with optimised HMM-GMM and dictionary. This transcription provides labels for DNN training. The DNN is trained to estimate HMM posterior states by minimizing the cross entropy loss L with l_1 regularization using back propagation:

$$\arg\min_{W} \sum_{i,j} L(\mathbf{x}_j^i, y_j^i, W) + \rho \|W\|_1 \qquad (5)$$

where $\mathbf{x}_j^i \in X_i$ is the jth feature vector of the ith utterance, y_j^i is the corresponding label and W is the set of network parameters, respectively. ρ is a constant parameter which is set to 10^{-6} in this work.

3. Experiments

We conducted several sets of experiments on the TIMIT corpus [19]. The TIMIT corpus provides a manually prepared dictionary and phone-level transcriptions with 61 phones. As a baseline, 61 phone models were trained using the TIMIT dictionary and the provided transcriptions. We used 12 mel frequency cepstral coefficients (MFCCs) and energy with their deltas and delta-deltas as descriptors of the acoustic space. The speech data was analyzed using a 25 ms Hamming window with a 10 ms frame shift. We evaluated phone based DNN-HMM, GMM-HMM and AAE based GMM-HMM model as baselines. The DNN architecture was comprised of 7 hidden layers. The first hidden layer had 2048 nodes, next 5 layers had 1024 nodes and the number of nodes at the last layer was equal to the number of HMM states to be predicted. All hidden layers were equipped with the Rectified Linear Unit (ReLU) non-linearity [20]. The input to the network was 11 contiguous frames of MFCCs. The networks were trained using mini-batch gradient descent based on back propagation with momentum. We applied dropout [16] to all hidden layers with dropout probability 0.5. The batch size was set to 128. HMMs had left-to-right, no-skipping topology with three states for each phoneme as opposed to one state for each AAE. HMMs were trained using a modified version of HTK [21] and DNNs were implemented using Lasagne [22].

3.1. Isolated Word Recognition

The first set of experiments were on the isolated word recognition to test the performance of the proposed methods and investigate the effects of hyper parameters such as the number of mixture components and the number of AAEs. For joint pronunciation estimation and acoustic models training, we collected a pronunciation training set comprising of words with more than 10 utterances from the TIMIT training set. The total number of utterances in the pronunciation training set was 12800. After excluding words with less than 4 characters (e.g., a and the), 339 distinct words were collected from the TIMIT test set for the isolated word speech recognition task, resulting in 3900 utterances in total. The baseline GMM based phone models were trained with 32 mixture components. During the GMM based AAE model training the number of mixtures was doubled for each iteration until it reached 128 mixtures as described in Section 2.4.

3.1.1. Comparison with phonetic approach

The word error rates (WER) of each method are shown in Table 1. The results show that the proposed data-driven method clearly outperforms the baseline methods. The proposed AAE-DNN method achieved 10.3% and 2.4% improvement over GMM and DNN based phonetic acoustic models, respectively. This suggests that a more accurate dictionary and better acoustic models can be obtained directly from training data without any human expertise. Moreover, AAE-DNN method improves the performance by 3.2% over the AAE-GMM method. This indicates that the DNN was successfully trained in the semi-supervised manner and the final model could effectively use the its expressive modeling power.

3.1.2. Number of AAEs

Our second experiment focused on the effects of the number of AAEs, i.e. N. We trained the dictionary and AAE models with $N = 64, 128, 192, 256, 320, 384, 448$. The word error rates of DNN and GMM based AAE models are illustrated in Figure 2.

Table 1: *Comparison of word error rates of each method on 339 words isolated word recognition (%). Baseline phone models are trained by using the TIMIT dictionary.*

Method	WER
Phone GMM	18.18
Phone DNN	10.31
AAE GMM	11.15
AAE DNN	**7.93**

Table 2: *Word error rates in % of AAE based recognizers with different number of AAEs and GMM mixture. The best performance for each number of AAE is plotted in Figure 2.*

# of AAE	# of mixture			
	16	32	64	128
64	19.48	18.33	17.52	**16.93**
128	14.33	13.87	**13.09**	13.70
192	13.39	13.31	**12.68**	13.98
256	11.97	**11.56**	12.65	14.33
320	11.46	**11.15**	11.69	14.10
384	11.63	**11.56**	12.14	13.75
448	**11.20**	11.33	12.45	-

The number of mixtures of the GMMs were determined experimentally as shown in Table 2. For DNN based AAE models, the best result are obtained with 384 AAEs in contrast to with 320 AAEs for the GMM based models. Interestingly, the optimal number of AAE states is far higher than the number of states of the phone models (61 phonemes $\times$ 3 states = 183 states). This is an indication that the proposed data-driven approach to jointly generate the sub-word units and dictionary models the acoustic space more precisely than the linguistically motivated phonetic units and the manually designed dictionary. It is also worthwhile to mention that the optimal number of DNN based AAE models was higher than that of GMM based models. This is perhaps due to the fact that the DNN was trained discriminatively, allowing to efficiently model the interaction between higher number of AAEs.

3.2. Continuous Speech Recognition

Unlike phoneme based ASRs, the proposed AAE based approach does not depend on linguistic knowledge. It is therefore interesting to compare these approaches on a real-world continues speech recognition (CSR) task. For this purpose, we used the SX records of the TIMIT corpus which contains 450 sentences spoken by 7 speakers, i.e. 3150 utterances in total. We prepared the test set by randomly selecting and putting aside one speaker for each sentence from the SX recordings and used the remaining samples as the training set (450 sentences $\times$ 6 speaker = 2700 utterances). We also included the SA and SI recordings of the TIMIT corpus in the training set. The number of AAEs was 384. The number of mixture components in the GMM based phone models was 64. The performance was evaluated in two scenarios: with and without language model. The language model employed in the baseline and the proposed methods is a simple bigram model.

Table 3 shows that the proposed AAE-DNN based approach significantly outperforms baseline methods in both scenarios. The performance improvements over the phone based HMM-DNN method in with and without the language model scenarios were 10.68% and 5.11%, respectively. The results suggest that the proposed data-driven dictionary and the AAE models are also useful for CSR and a more accurate representation of speech signals can be learned automatically. We observed that all 384 AAEs were actually used in the trained dictionary, and the dictionary tend to assign 39% more HMM states on average to each word as compare with the TIMIT phonetic dictionary. This means that in AAEs, the stay-in-state probability is smaller resulting in more frequent state transitions. This suggests that by using AAEs, the acoustic space was modeled at a higher resolution. This consequently increased the precision of the word pronunciations.

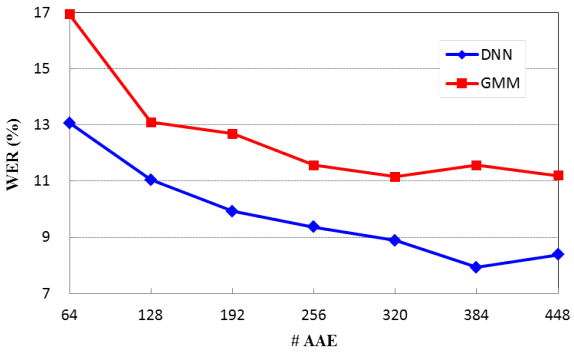

Figure 2: *Performance of AAE based recognizers with different number of AAEs on test set with 339 words.*

Table 3: *Comparison of word error rate of each method on continuous speech recognition. In column No LM, no language model was used.*

Method	No LM	Bigram
Phone GMM	71.11	43.54
Phone DNN	50.18	20.89
AAE GMM	59.52	32.36
AAE DNN	**39.05**	**15.78**

4. Conclusions

In this work we proposed a novel joint dictionary and sub-word unit learning framework for ASRs. The proposed method does not require linguistic expertise, and can automatically create the set of sub-word units and the corresponding pronunciation dictionary. In our method, reliable pronunciations are estimated from multiple utterances by an efficient approximation of K-dimensional Viterbi algorithm which estimates the most probable HMM state sequence common to multiple utterances of a word. Experimental results show that the proposed method significantly outperforms the phone based methods which even get manually prepared dictionary and hand crafted transcriptions as inputs. We further investigated the effects of the number of data-driven sub-word units and showed that the optimal number of sub-word units is much higher than the total number of HMM states of the 61 phones. The future works will be directed towards applying the proposed method to speech recognition for under-resourced languages and large vocabulary continuous speech recognition tasks.

5. References

[1] A. Das and M. Hasegawa-Johnson, "Cross-lingual transfer learning during supervised training in low resource scenarios," in *Proc. Interspeech*, 2015, pp. 1–5.

[2] Y. Qian, D. Povey, and J. Liu, "State-level data borrowing for low-resource speech recognition based on subspace GMMs," in *Proc. Interspeech*, 2011, pp. 553–556.

[3] L. Besacier, E. Barnard, A. Karpov, and T. Schultz, "Automatic speech recognition for under-resourced languages : A survey," *Speech Communication*, vol. 56, pp. 85–100, 2014.

[4] M. Saraçlar, H. Nock, and S. Khudanpur, "Pronunciation modeling by sharing Gaussian densities across phonetic models," *Computer Speech & Language*, vol. 14, no. 2, pp. 137–160, 2000.

[5] M. Wester, "Pronunciation modeling for ASR - Knowledge-based and data-derived methods," *Computer Speech and Language*, vol. 17, no. 1, pp. 69–85, 2003.

[6] T. Hain, "Implicit modelling of pronunciation variation in automatic speech recognition," *Speech Communication*, vol. 46, no. 2, pp. 171–188, 2005.

[7] I. Mcgraw, I. Badr, and J. R. Glass, "Learning lexicons from speech using a pronunciation mixture model," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 21, no. 2, pp. 357–366, 2013.

[8] R. Singh, B. Raj, and R. M. Stern, "Automatic generation of subword units for speech recognition systems," *IEEE Transactions on Speech and Audio Processing*, vol. 10, no. 2, pp. 89–99, 2002.

[9] A. Ghoshal, D. Povey, M. Agarwal, P. Akyazi, N. Goel, M. Karafi, A. Rastrow, R. C. Rose, P. Schwarz, S. Thomas, and I. Allahabad, "Approaches to automatic lexicon learning with limited trainging examples," in *Proc. ICASSP*, 2010, pp. 5094–5097.

[10] T. Naghibi, S. Hoffmann, and B. Pfister, "An efficient method to estimate pronunciation from multiple utterances," in *Proc.Interspeech*, no. August, 2013, pp. 1951–1955.

[11] T. Holter and T. Svendsen, "Combined optimisation of baseforms and model parameters in speech recognition based on acoustic subword units," in *IEEE Workshop Automatic Speech Recognition*, 1997, pp. 199–206.

[12] M. Bacchiani and M. Ostendorf, "Joint lexicon, acoustic unit inventory and model design," *Speech Communication*, vol. 29, no. 2, pp. 99–114, 1999.

[13] T. Svendsen, "Pronunciation modeling for speech technology," in *International Conference on Signal Processing and Communications (SPCOM)*, 2004, pp. 11–16.

[14] H. Mokbel and D. Jouvet, "Derivation of the optimal set of phonetic transcriptions for a word from its acoustic realizations," in *Speech Communication*, vol. 29, no. 1, 1999, pp. 49–64.

[15] M. Gerber, T. Kaufmann, and B. Pfister, "Extended Viterbi algorithm for optimized word HMMs," in *ICASSP*, 2011, pp. 4932–4935.

[16] G. Hinton, L. Deng, D. Yu, G. Dahl, A. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. Sainath, and B. Kingsbury, "Deep neural networks for acoustic modeling in speech recognition," *Signal Processing Magazine*, 2012.

[17] O. Abdel-Hamid, A. Mohamed, H. Jiang, and G. Penn, "Applying convolutional neural networks concepts to hybrid NN-HMM model for Speech Recognition," in *ICASSP*, 2012, pp. 4277–4280.

[18] Y. Linde, A. Buzo, and R. M. Gray, "An algorithm for vector quantizer design," *IEEE Transactions on Communications*, vol. 28, no. 1, pp. 84–95, 1980.

[19] W. Fisher, G. Doddington, and K. Goudie-Marshall, "The DARPA speech recognition research database: Specifications and status," in *Proc. DARPA Workshop on Speech Recognition*, 1986, pp. 93–99.

[20] G. E. Dahl, T. N. Sainath, and G. E. Hinton, "Improving deep neural networks for LVCSR using rectified linear units and dropout," in *Proc. ICASSP*, 2013, pp. 8609–8613.

[21] S. Young, G. Evermann, M. Gales, T. Hain, D. Kershaw, X. Liu, G. Moore, J. Odell, D. Ollason, D. Povey, V. Valtchev, and P. Woodland, "The HTK Book (for HTK Version 3.4.1)," http://htk.eng.cam.ac.uk, 2009, University of Cambridge, UK.

[22] E. Battenberg, S. Dieleman, D. Nouri, E. Olson, C. Raffel, J. Schlüter, S. K. Sønderby, D. Maturana, M. Thoma *et al.*, "Lasagne: First release." http://dx.doi.org/10.5281/zenodo.27878, Aug. 2015.