Joint Phoneme Segmentation Inference and Classification using CRFs

Dimitri Palaz*†, Mathew Magimai-Doss* and Ronan Collobert*
*Idiap Research Institute, Martigny, Switzerland
†Ecole Polytechnique Fédérale de Lausanne (EPFL), Lausanne, Switzerland
{dimitri.palaz, mathew}@idiap.ch, ronan@collobert.com

***Abstract*—State-of-the-art phoneme sequence recognition systems are based on hybrid hidden Markov model/artificial neural networks (HMM/ANN) framework. In this framework, the local classifier, ANN, is typically trained using Viterbi expectation-maximization algorithm, which involves two separate steps: phoneme sequence segmentation and training of ANN. In this paper, we propose a CRF based phoneme sequence recognition approach that simultaneously infers the phoneme segmentation and classifies the phoneme sequence. More specifically, the phoneme sequence recognition system consists of a local classifier ANN followed by a conditional random field (CRF) whose parameters are trained jointly, using a cost function that discriminates the true phoneme sequence against all competing sequences. In order to efficiently train such a system we introduce a novel CRF based segmentation using acyclic graph. We study the viability of the proposed approach on TIMIT phoneme recognition task. Our studies show that the proposed approach is capable of achieving performance similar to standard hybrid HMM/ANN and ANN/CRF systems where the ANN is trained with manual segmentation.**

Index Terms: phoneme classification, phonetic segmentation, conditional random fields, convolutional neural network.

I. Introduction

State-of-the-art ASR systems are based on Hidden Markov Models (HMM) which divide the sequence recognition task into sub-tasks: feature extraction, modeling individual units or sub-units of the sequence and decoding the whole sequence. In this framework, the modeling part is performed by a local classifier, which can be a generative model, such as Gaussian Mixture Models (GMM), or a discriminative model, such as Artificial Neural Networks (ANN). At each time step or frame, these models perform a local classification of individual units or sub-units, typically phonemes, of the speech sequence, which is then decoded. More recently, it has been shown that ANNs with deep architectures can achieve better system when compared to GMMs [1]–[4]. Neural networks are supervised classifiers. In order to train them a class label for each time step is required. Often, the ASR system training process has access to the class label sequence (or transcription) corresponding to the speech utterance or feature sequence but not the segmentation (or alignment). Thus, Viterbi expectation maximization approach is typically employed, which splits the ANN training process into two iterative steps: estimating the segmentation for the whole training data and estimating the parameters of the ANN.

Recent advances in Machine Learning have made possible systems that can be trained in an end-to-end manner, i.e. systems in which every step is learned simultaneously. These models are usually trained in a discriminative fashion, leading to globally optimized systems. It can be referred to as *deep learning*. Such systems have been proposed in natural language processing [5] or image recognition [6]. In speech recognition, early works have investigated global training of hybrid HMM/ANN systems [7]. Recently, ANN/CRF based systems have been proposed [8]–[10]. End-to-end trained systems have also been proposed [11], [12]. However, in these studies it was always assumed that the segmentation of the training data is provided.

The present paper proposes an approach that *simultaneously* infers phoneme segmentation and classification. The core of our system consists of a local ANN classifier followed by a conditional random field (CRF) [13] decoder. In contrast with previous approaches, we train our CRF such that it learns the most likely phoneme segmentation given the phoneme transcription sequence. In this framework, following the Graph Transformed Network (GTN) [14] approach, the CRF segments and back-propagates error gradient during training computed by discriminating the true sequence against competing sequences, to the local classifier. In order to do this efficiently, we introduce a novel CRF decoding scheme. If L denotes the phoneme transcription of an utterance and X denotes the corresponding input feature sequence, the proposed approach can be seen as a sequence classification system based on maximization of $P(L|X)$ rather than maximization of $P(L, X)$ as in the case of standard HMM-based approach. We evaluate the performance of the proposed scheme on the TIMIT database. We compare two systems: The first one is an artificial neural network composed of many hidden layers. The second one is based on Convolutional Neural Networks [15] (CNNs). Both systems take MFCC features as input. We show that both systems achieve performance similar to their counterparts where a hand-labeled segmentation is provided.

The remainder of the paper is organized as follows. Section II presents the related work. Section III presents the proposed framework. Section IV and Section V present the experimental setup and the results, respectively. Finally, Section VI concludes the paper.

II. Related Work

In the hybrid HMM/ANN framework [16], the phoneme sequence recognition is performed in two steps: (1) each frame is modeled by a neural network and (2) the sequence is decoded using HMM. However, most speech corpora are labeled only at the word level. To retrieve phoneme-level labels, a dictionary is used, which expresses words as a sequence of phonemes. Having the phoneme sequence for the speech waveform is not sufficient for training a neural network model, because the phoneme segmentation is not known. Manual segmentation can be quite precise, but is costly and time consuming, and is practically unfeasible for large datasets.

The other approach to obtain a segmentation suitable to train an ANN is the *Viterbi EM* algorithm [17]. It consists of two iterative steps: (1) Expectation (or E-step), which finds the best segmentation maximizing the likelihood of the joint probability distribution $P(L, X)$, with X being the input sequence, and (2) Maximization (or M-step), where an ANN with a cost function based on local classification error, such as cross-entropy, is trained. In this approach, at each M-step, a new neural network has to be trained from scratch, which requires each time several epochs of training. In that respect, this approach can be time consuming for large databases. Instead, the common approach is to train a HMM/GMM system to obtain a segmentation and then train an ANN afterwards. Thereby, the segmentation and the classifier training is done independently.

In contrast to the local classification approach of hybrid systems, sequence-level classification criteria have been proposed. In this approach, a local classifier, usually a neural network, is first trained with cross-entropy criterion. Then, the network is trained in a sequence-discriminative framework, using criteria inspired from HMM/GMM systems [18], like Maximum Mutual Information (MMI), state Minimum Bayesian Risk (sMBR) or Minimum Phone Error (MPE). These systems have been shown to improve performance compared to frame-level training [19]–[21]. Sequential deep neural networks have also been proposed [22] using the deep belief network pre-training framework.

III. Proposed framework

Hybrid HMM/ANN-based phoneme sequence recognition systems are locally discriminative but globally generative (training and recognition performed by maximizing $P(L, X)$). In this section, we propose a phoneme sequence recognition system that is discriminative both locally and globally. More precisely, the proposed system consists of a local ANN classifier followed by a CRF. In this approach, unlike hybrid HMM/ANN systems, segmentation (CRF) and phoneme prediction (ANN) are trained jointly by maximizing $P(L|X)$. As described in detail in the remainder of the present section, the training of the proposed system is a particular case of the forward training of graph transformer network, where the segmentation is first obtained through a simple CRF decoding scheme and the gradient is then back-propagated to the local classifier.

A. Model

We consider a simple CRF, where we define a graph (see Figure 1) with nodes for each frame in the input sequence, and for each label. Transition scores, denoted as a matrix A, are assigned to the edges between phonemes, and network prediction scores $f(\cdot)$ are assigned to the nodes. This CRF allows to discriminatively train a simple duration model over the network output scores. Given an input data sequence $\{x_1, \ldots, x_T\} = [x]_1^T$ and a label path $\{i_1, \ldots, i_T\} = [i]_1^T$ on the graph, a score for the path can be defined:

$$s([x]_1^T, [i]_1^T, \theta) = \sum_{t=1}^{T} \left(f_{i_t}(x_t) + A_{i_t, i_{t-1}} \right) \tag{1}$$

with θ being the network parameters and the matrix A.

Compared to classical CRFs, this model is a non-linear CRF, as $f(\cdot)$ is the output of a non-linear network. At inference time, given a input sequence $[x]_1^T$, the best label path $[i^\star]_1^T$ can be found by minimizing (1), more precisely by using the Viterbi algorithm.

$$[i^\star]_1^T = \underset{[j]_1^T}{\operatorname{argmax}} (s([x]_1^T, [j]_1^T, \theta)) \,. \tag{2}$$

Note that this sequence of tags assigns a label for each *frame* in the given input. Given that a phoneme can last several frames, the final phoneme sequence prediction is obtained by aggregating successive identical phoneme tags in $[i^\star]_1^T$.

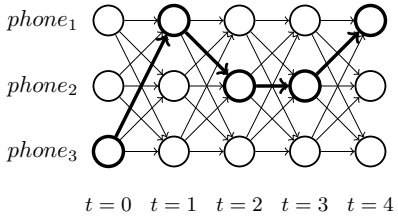

Fig. 1. Illustration of the CRF graph.

B. Joint segmentation and training

Given a sequence of phoneme labels $\{l_1, l_2 ..., l_N\} = [l]_1^N$, and a data sequence $[x]_1^T$, the problem of segmentation consists in finding a sequence $[i]_1^T$ (over T frames) of labels, such that after aggregation of successive identical labels in $[i]_1^T$, one matches the sequence $[l]_1^N$. In the classical ANN/HMM or HMM/GMM framework, $[i]_1^T$ is assumed to be known. We consider the setup where only the phoneme labels sequence $[l]_1^N$ is given. To infer the segmentation, we need to constrain the CRF graph such that it covers all possible sequences $[i]_1^T$ that could match $[l]_1^N$ after label aggregation.

1) Segmentation graph: The constraints over time imposed by the label sequence $[l]_1^N$ can be written as a directed cyclic graph, where each node represents one label from the sequence, as illustrated in Figure 2. At every time step, the path can either stay in the current node through the loop or go to the next node (or label).

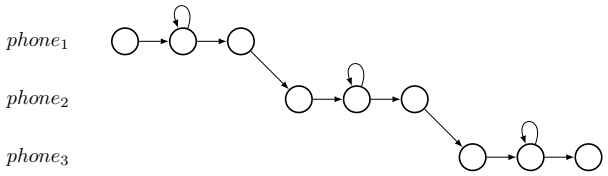

Fig. 2. Illustration of the cyclic graph for 3 classes.

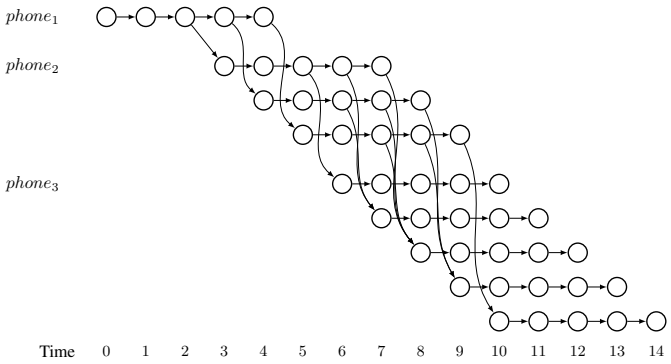

Fig. 3. Illustration of the acyclic expanded graph for 3 classes, with $t_{min} = 3$ and $t_{max} = 5$

In order to implement such graph, we need to expand it to an acyclic graph over a sequence duration of T frames. We introduce two parameters, t_{min} and t_{max}, which represent the minimum and maximum time the path can stay in the same label. To enforce these conditions, the acyclic graph must have multiple parallel branches for each label. All parallel branches of the same label share their weights, i.e. $f_t(x_t)$ is the same for each time t in each parallel branch. An illustration is provided in Figure 3.

2) Training procedure: In the following, we denote the unconstrained CRF graph over T frames as $\mathcal{U}_T$ (Figure 1), and we also denote the graph constrained to the right sequence of labels $[l]_1^N$ as $\mathcal{C}_T$ (Figure 3). Assuming for now there is only one unique sequence $[i]_1^T$ over the T frames corresponding to the sequence of provided phoneme labels $[l]_1^N$ (e.g. if a manual segmentation is provided), in a classical CRF framework one would maximize the conditional log-likelihood $\mathcal{L}(\theta)$, given by:

$$\mathcal{L}(\theta) = \sum_{n=1}^{N} \log(p([i_n]_1^T | [x_n]_1^T, \theta)) \tag{3}$$

for each input speech sequence $[x_n]_1^T$ and label sequence $[i_n]_1^T$ over the whole training set. In a standard CRF setup, scores $s([x]_1^T, [i]_1^T)$ are interpreted as a conditional probability $p([i]_1^T | [x]_1^T, \theta)$ by taking them to the exponential (such that there are positive) and normalizing them over all possible label paths $[j]_1^T$:

$$\log(p([i]_1^T | [x]_1^T, \theta)) = s([x]_1^T, [i]_1^T, \theta) - \underset{[j]_1^T \in \mathcal{U}_T}{\text{logadd}}\, s([x]_1^T, [j]_1^T, \theta)\,, \tag{4}$$

where the logadd operation is defined for simplification purpose as:

$$\underset{i}{\text{logadd}}(z_i) = \log(\sum_i e^{z_i})\,. \tag{5}$$

While the number of terms in the logadd operation grows exponentially with the length of the input sequence, the Forward recursive algorithm can be applied to compute this quantity efficiently.

Our setup is however more general, as we look for the best sequence $[i^\star]_1^T$ ($[i^\star]_1^T \subset \mathcal{C}_T$) matching the right sequence of labels $[l]_1^N$, as shown in Figure 3. Finding this sequence corresponds to solve the maximization problem

$$\max_{[j]_1^T \in \mathcal{C}_T} s([x]_1^T, [j]_1^T, \theta). \tag{6}$$

This is achieved with a Viterbi algorithm, as in (2). Integrating this best path into (4) leads to the following likelihood:

$$\mathcal{L}(\theta) = \max_{[j]_1^T \in \mathcal{C}_T} s([x]_1^T, [j]_1^T, \theta) - \underset{[j]_1^T \in \mathcal{U}_T}{\text{logadd}}\, s([x]_1^T, [j]_1^T, \theta)\,. \tag{7}$$

We use the stochastic gradient ascent algorithm [23] to train our complete architecture. The gradient is back-propagated through the Forward recursion.

IV. Experimental Setup

A. Architectures

In this paper, we focus on the joint segmentation and classification training. To evaluate the proposed approach, we investigate two systems for acoustic modeling: artificial neural networks (ANN) and convolutional neural networks (CNN). The ANN system is composed of several hidden layers. For the CNN system, the proposed architecture is composed of several filtering stages, implemented by convolutional layers and of a classification stage, implemented by hidden layers. More details on this architecture can be found in [24].

The TIMIT acoustic-phonetic corpus was selected for the experiments. It consists of 3,696 training utterances (sampled at 16kHz) from 462 speakers, excluding the SA sentences. The cross-validation set consists of 400 utterances from 50 speakers. The core test set was used to report the results. It contains 192 utterances from 24 speakers, excluding the validation set. The phoneme set is composed of 39 phonemes, as presented in [25]. A phoneme segmentation is provided with this corpus. We refer to this segmentation as "manual segmentation".

The models were trained on MFCC features. They were computed (with HTK [26]) using a 25 ms Hamming window on the speech signal, with a shift of 10 ms. The signal is represented using 13th-order coefficients along with their first and second derivatives, computed on a 9-frame context, resulting in 39 dimensional vector. Both systems are trained following the procedure presented in Section III, with parameters update every sequence. Also, we do not use any pre-training scheme, i.e. the network is initialized randomly.

B. Hyper-parameters

The hyper-parameters of the segmentation graph are the labeled frame duration and the minimum and maximum phoneme duration t_{min} and t_{max}. To allow comparison with manual segmentation, the frame duration was set to 10ms.

The minimum duration t_{min} was set to 30ms, or 3 frames. The maximum duration t_{max} was set to 200ms, or 20 frames. Early stopping on the cross-validation set was used to determine the hyper-parameters of the models. For the ANN systems, the hyper-parameters are the context frames and the width of hidden layers. 90 ms of context and 500 nodes for each hidden layer were found. For the CNN system, the hyper-parameters are: the input window size, the kernel width kW and shift dW of the convolutions, the number of filters d_{out} and the width of the hidden layer. The best performance was found with: 3 convolutional layers, 250 ms of context, 5 frames kernel width, 1 frames shift, 80 filters and 500 hidden units. The experiments were implemented using the *torch7* toolbox [27].

C. Baseline systems

The proposed approach is compared to two baseline systems. The first one is a hybrid HMM/ANN system from our previous work [24]. The classifier is a three-layer ANN, trained using the cross-entropy criterion. The decoding of the sequence was performed by a standard HMM decoder, with 3 states minimum duration constraint, and considering all phonemes equally probable. This system is referred to as "HMM/ANN". We also compare our approach to the CRF based system proposed in [8]. This system uses local posterior estimates provided by an ANN (trained separately) as features for the CRF. This system is referred as "ANN/CRF". All these systems are trained using manual segmentation provided with the database.

V. Results and discussion

We first evaluate the ANN based system trained using manual segmentation. The system is trained by minimizing the likelihood in Equation (4). For comparison, we also train it using the cross-entropy criterion, as presented in [24]. The results are presented in Table I, expressed in terms of phoneme error rate (PER), along with baseline systems. We report the mean and standard deviation, computed on 10 experiments. It can be observed that the proposed ANN-based system yields performance similar to the baselines. Increasing the number of hidden layers shows a slight improvement. Finally, the system trained under the proposed criterion slightly outperforms the one trained under the cross-entropy criterion. These results are also comparable with the system proposed in [28], which uses the same training criterion. This system achieves 31.8 % PER on TIMIT using a single hidden layer with 1000 units, which is larger than the 500 units used in our experiments.

Table II presents the performance of the two proposed systems (ANN and CNN) trained under manual and learned segmentation conditions. The CNN-based system trained using the proposed approach is able to outperform the ANN-based system. More importantly, the proposed approach using learned segmentation yields similar performance using manual segmentation. The time for training one sentence using learned segmentation is 800 ms on average. This is between 5 and 20 times slower than the training using manual segmentation. The computation time is the same for the inference step.

These results clearly indicate that the proposed joint training approach, which maximizes $P(L|X)$, can be a good alternative to the independent training approach, based on maximizing $P(L, X)$.

TABLE I
Evaluation of the proposed training scheme on TIMIT core testset using manual segmentation. Results are in PER.

Systems	# Hidden Layers	Training criterion	
		Cross-entropy	Proposed
Previous Works			
ANN/CRF [8]	1	33.3	-
HMM/ANN [24]	1	33.4	-
Proposed approach			
ANN	1	34.4 $\pm$ 0.4	33.1 $\pm$ 0.5
ANN	2	32.9 $\pm$ 0.5	32.5 $\pm$ 0.7
ANN	3	32.8 $\pm$ 0.4	31.7 $\pm$ 0.4

TABLE II
Recognition performance on TIMIT core testset using manual and learned segmentation.

System	# Hidden Layers	PER [%]	
		Manual	Learned
ANN	1	33.4	33.4
ANN	3	31.8	31.7
CNN	3+1	28.4	28.4

In the proposed approach, the models are trained by emphasizing the score of the true sequence while de-emphasizing the score of all other sequences. The proposed cost function can be compared to the Maximum Mutual Information (MMI) criterion proposed in the sequence-discriminative training framework [21]. However, the key difference is the score normalization. For the MMI case, the normalization is a sum over all possible word hypotheses, which is practically infeasible to estimate. So it is approximated by decoding the training data using an unigram language model and generating a lattice. In the proposed criterion, it is computed exactly by using a fully connected phone model. This is the most relaxed model one can have, as it includes every possible word sequence.

VI. Conclusion

In this paper, we proposed an approach that simultaneously infers the phoneme segmentation and learns the phoneme classification in an end-to-end manner. To efficiently train the system, we introduced a novel CRF-based segmentation scheme, based on an acyclic graph. The proposed system yields similar results to the hybrid HMM/ANN baseline, using a manual segmentation. For future work, we will investigate the proposed approach for discriminative ASR acoustic modeling. We will also investigate using raw speech signal as input. Finally, we will investigate graphemes as sequence sub-units.

Acknowledgement

This work was supported by the HASLER foundation (www.haslerstiftung.ch) through the grant "Universal Spoken Term Detection with Deep Learning" (DeepSTD).

REFERENCES

[1] H. Lee, P. Pham, Y. Largman, and A. Y. Ng, "Unsupervised feature learning for audio classification using convolutional deep belief networks," in *Advances in Neural Information Processing Systems 22*, 2009, pp. 1096–1104.

[2] F. Seide, G. Li, and D. Yu, "Conversational speech transcription using context-dependent deep neural networks," in *Proc. Interspeech*, 2011, pp. 437–440.

[3] G. Hinton, L. Deng, D. Yu, G. E. Dahl, A. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, and T. N. Sainath, "Deep neural networks for acoustic modeling in speech recognition: the shared views of four research groups," *Signal Processing Magazine, IEEE*, vol. 29, no. 6, p. 8297, 2012.

[4] A. Mohamed, G. Dahl, and G. Hinton, "Acoustic modeling using deep belief networks," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 20, no. 1, pp. 14 –22, jan. 2012.

[5] R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, and P. Kuksa, "Natural language processing (almost) from scratch," *The Journal of Machine Learning Research*, vol. 12, pp. 2493–2537, 2011.

[6] A. Krizhevsky, I. Sutskever, and G. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems 25*, 2012, pp. 1106–1114.

[7] Y. Bengio, R. De Mori, G. Flammia, and R. Kompe, "Global optimization of a neural network-hidden markov model hybrid," in *Neural Networks, 1991., IJCNN-91-Seattle International Joint Conference on*, vol. ii, jul 1991, pp. 789 –794 vol.2.

[8] J. Morris and E. Fosler-Lussier, "Conditional random fields for integrating local discriminative classifiers," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 16, no. 3, pp. 617–628, Mar. 2008.

[9] A. Mohamed, D. Yu, and L. Deng, "Investigation of full-sequence training of deep belief networks for speech recognition," in *Proc. Interspeech*, vol. 10, 2010, pp. 2846–2849.

[10] Y. Kubo, T. Hori, and A. Nakamura, "Integrating deep neural networks into structural classification approach based on weighted finite-state transducers." in *INTERSPEECH*, 2012.

[11] A. Graves, *Sequence transduction with recurrent neural networks*. Springer, 2012, vol. 385.

[12] D. Palaz, R. Collobert, and M. Magimai. -Doss, "End-to-end Phoneme Sequence Recognition using Convolutional Neural Networks," *ArXiv e-prints*, Dec. 2013.

[13] J. Lafferty, A. McCallum, and F. Pereira, "Conditional random fields: Probabilistic models for segmenting and labeling sequence data," 2001.

[14] L. Bottou, Y. Bengio, and Y. LeCun, "Global training of document processing systems using graph transformer networks." in *In Proc. of Computer Vision and Pattern Recognition*. Puerto-Rico., 1997, pp. 490–494.

[15] Y. LeCun, "Generalization and network design strategies," in *Connectionism in Perspective*, R. Pfeifer, Z. Schreter, F. Fogelman, and L. Steels, Eds. Zurich, Switzerland: Elsevier, 1989.

[16] N. Morgan and H. Bourlard, "Continuous speech recognition," *Signal Processing Magazine, IEEE*, vol. 12, no. 3, pp. 24 –42, May 1995.

[17] L. Rabiner and B.-H. Juang, *Fundamentals of Speech Recognition*. Prentice-Hall, Inc., 1993.

[18] M. Gales and S. Young, "The application of hidden markov models in speech recognition," *Found. Trends Signal Process.*, vol. 1, no. 3, pp. 195–304, Jan. 2007.

[19] B. Kingsbury, "Lattice-based optimization of sequence classification criteria for neural-network acoustic modeling," in *Proc. of ICASSP*. IEEE, 2009, pp. 3761–3764.

[20] W. Guangsen and K. C. Sim, "Sequential classification criteria for NNs in automatic speech recognition," 2011.

[21] K. Vesel, A. Ghoshal, L. Burget, and D. Povey, "Sequence-discriminative training of deep neural networks," in *Proc. INTERSPEECH*, 2013, pp. 2345–2349.

[22] G. Andrew and J. Bilmes, "Backpropagation in sequential deep neural networks," in *NIPS*, 2013.

[23] L. Bottou, "Stochastic gradient learning in neural networks," in *Proceedings of Neuro-Nmes 91*. Nimes, France: EC2, 1991.

[24] D. Palaz, R. Collobert, and M. Magimai.-Doss, "Estimating phoneme class conditional probabilities from raw speech signal using convolutional neural networks," in *Proc. Interspeech*, 2013.

[25] K. F. Lee and H. W. Hon, "Speaker-independent phone recognition using hidden markov models," *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 37, no. 11, pp. 1641–1648, 1989.

[26] S. Young, G. Evermann, D. Kershaw, G. Moore, J. Odell, D. Ollason, V. Valtchev, and P. Woodland, "The htk book," *Cambridge University Engineering Department*, vol. 3, 2002.

[27] R. Collobert, K. Kavukcuoglu, and C. Farabet, "Torch7: A matlab-like environment for machine learning," in *BigLearn, NIPS Workshop*, 2011.

[28] R. Prabhavalkar and E. Fosler-Lussier, "Backpropagation training for multilayer conditional random field based phone recognition," in *Proc. of ICASSP*. IEEE, 2010, pp. 5534–5537.