Segmental Recurrent Neural Networks for End-to-end Speech Recognition

Segmental Recurrent Neural Networks for End-to-end Speech Recognition

Liang Lu[1*], *Lingpeng Kong*[2*], *Chris Dyer*[2], *Noah A. Smith*[3], *and Steve Renals*[1]

[1]Centre for Speech Technology Research, The University of Edinburgh, Edinburgh, UK
[2]School of Computer Science, Carnegie Mellon University, Pittsburgh, USA
[3]Computer Science & Engineering, The University of Washington, Seattle, USA
{liang.lu, s.renals}@ed.ac.uk, {lingpenk, cdyer}@cs.cmu.edu, nasmith@cs.washington.edu

Abstract

We study the segmental recurrent neural network for end-to-end acoustic modelling. This model connects the segmental conditional random field (CRF) with a recurrent neural network (RNN) used for feature extraction. Compared to most previous CRF-based acoustic models, it does not rely on an external system to provide features or segmentation boundaries. Instead, this model marginalises out all the possible segmentations, and features are extracted from the RNN trained together with the segmental CRF. In essence, this model is self-contained and can be trained end-to-end. In this paper, we discuss practical training and decoding issues as well as the method to speed up the training in the context of speech recognition. We performed experiments on the TIMIT dataset. We achieved 17.3% phone error rate (PER) from the first-pass decoding — the best reported result using CRFs, despite the fact that we only used a zeroth-order CRF and without using any language model.

Index Terms: end-to-end speech recognition, segmental CRF, recurrent neural networks.

1. Introduction

Speech recognition is a typical sequence to sequence transduction problem, i.e., given a sequence of acoustic observations, the speech recognition engine decodes the corresponding sequence of words (or phonemes). A key component in a speech recognition system is the acoustic model, which computes the conditional probability of the output sequence given the input sequence. However, directly computing this conditional probability is challenging due to many factors including the variable lengths of the input and output sequences. The hidden Markov model (HMM) converts this sequence-level classification task into a frame-level classification problem, where each acoustic frame is classified into one of the hidden states, and each output sequence corresponds to a sequence of hidden states. To make it computationally tractable, HMMs usually rely on the conditional independence assumption and the first-order Markov rule — the well-known weaknesses of HMMs [1]. Furthermore, the HMM-based pipeline is composed of a few relatively independent modules, which makes the joint optimisation nontrivial.

There has been a consistent research effort to seek architectures to replace HMMs and overcome their limitation for acoustic modelling, e.g., [2, 3, 4, 5]; however these approaches have not yet improved speech recognition accuracy over HMMs. In the past few years, several neural network based approaches have been proposed and demonstrated promising results. In particular, the connectionist temporal classification (CTC) [6, 7, 8, 9] approach defines the loss function directly to maximise the conditional probability of the output sequence given the input sequence, and it usually uses a recurrent neural network to extract features. However, CTC simplifies the sequence-level error function by a product of the frame-level error functions (i.e., independence assumption), which means it essentially still does frame-level classification. It also requires the lengths of the input and output sequence to be the same, which is inappropriate for speech recognition. CTC deals with this problem by replicating the output labels so that a consecutive frames may correspond to the same output label or a *blank* token.

Attention-based RNNs have been demonstrated to be a powerful alternative sequence-to-sequence transducer, e.g., in machine translation [10], and speech recognition [11, 12, 13]. A key difference of this model from HMMs and CTCs is that the attention-based approach does not apply the conditional independence assumption to the input sequence. Instead, it maps the variable-length input sequence into a fixed-size vector representation at each decoding step by an attention-based scheme (see [10] for further explanation). It then generates the output sequence using an RNN conditioned on the vector representation from the source sequence. The attentive scheme suits the machine translation task well, because there may be no clear alignment between the source and target sequence for many language pairs. However, this approach does not naturally apply to the speech recognition task, as each output token only corresponds to a small size window of acoustic spectrum.

In this paper, we study segmental RNNs [14] for acoustic modelling. This model is similar to CTC and attention-based RNN in the sense that an RNN encoder is also used for feature extraction, but it differs in the sense that the sequence-level conditional probability is defined using an segmental (semi-Markov) CRF [15], which is an extension on the standard CRF [16]. There have been numerous works on CRFs and their variants for speech recognition, e.g, [4, 5, 17] (see [18] for an overview). In particular, feed-forward neural networks have been used with segmental CRFs for speech recognition [19, 20]. However, segmental RNN is different in that it is an end-to-end model — it does not depend on an external system to provide segmentation boundaries and features, instead, this model is trained by marginalising out all possible segmentations, while the features are derived from the encoder RNN, which is trained jointly with the segmental CRF. Our experiments were performed on the TIMIT dataset, and we achieved 17.3% PER from first-pass decoding with zeroth-order CRF and without using any language model — the best reported result using CRFs.

* Equal contribution. Lu and Renals are funded by the UK EPSRC Programme Grant EP/I031022/1, Natural Speech Technology (NST). The NST research data collection may be accessed at http://datashare.is.ed.ac.uk/handle/10283/786.

2. Segmental Recurrent Neural Networks

2.1. Segmental Conditional Random Fields

Given a sequence of acoustic frames $\mathbf{X} = \{\mathbf{x}_1, \cdots, \mathbf{x}_T\}$ and its corresponding sequence of output labels $\mathbf{y} = \{y_1, \cdots, y_J\}$, where $T \geq J$, segmental (or semi-Markov) conditional random field defines the sequence-level conditional probability with the auxiliary segment labels $\mathbf{E} = \{\mathbf{e}_1, \cdots, \mathbf{e}_J\}$ as

$$P(\mathbf{y}, \mathbf{E} \mid \mathbf{X}) = \frac{1}{Z(\mathbf{X})} \prod_{j=1}^{J} \exp f\left(y_j, \mathbf{e}_j, \mathbf{X}\right), \tag{1}$$

where $\mathbf{e}_j = \langle s_j, n_j \rangle$ is a tuple of the beginning (s_j) and the end (n_j) time tag for the segment of y_j, and $n_j > s_j$ while $n_j, s_j \in [1, T]$; $y_j \in \mathcal{Y}$ and $\mathcal{Y}$ denotes the vocabulary set; $Z(\mathbf{X})$ is the normaliser that that sums over all the possible $(\mathbf{y}, \mathbf{E})$ pairs, i.e.,

$$Z(\mathbf{X}) = \sum_{\mathbf{y}, \mathbf{E}} \prod_{j=1}^{J} \exp f\left(y_j, \mathbf{e}_j, \mathbf{X}\right). \tag{2}$$

Here, we only consider the zeroth-order CRF, while the extension to higher order models is straightforward. Similar to other CRF-based models, the function $f(\cdot)$ is defined as

$$f\left(y_j, \mathbf{e}_j, \mathbf{X}\right) = \mathbf{w}^\top \Phi(y_j, \mathbf{e}_j, \mathbf{X}), \tag{3}$$

where $\Phi(\cdot)$ denotes the feature function, and $\mathbf{w}$ is the weight vector. Previous works on CRF-based acoustic models mainly use heuristically handcrafted feature function $\Phi(\cdot)$. They also usually rely on an external system to provide the segment labels. In this paper, we define $\Phi(\cdot)$ using neural networks, and the segmentation $\mathbf{E}$ is marginalised out during training, which makes our model self-contained.

2.2. Feature Representations

We use neural networks to define the feature function $\Phi(\cdot)$, which maps the acoustic segment and its corresponding label into a joint feature space. More specifically, y_j is firstly represented as a one-hot vector $\mathbf{v}_j$, and it is then mapped into a continuous space by a linear embedding matrix $\mathbf{M}$ as

$$\mathbf{u}_j = \mathbf{M}\mathbf{v}_j \tag{4}$$

Given the segment label $\mathbf{e}_j$, we use an RNN to map the acoustic segment to a fixed-dimensional vector representation, i.e.,

$$\mathbf{h}_1^j = r(\mathbf{h}_0, \mathbf{x}_{s_j}) \tag{5}$$

$$\mathbf{h}_2^j = r(\mathbf{h}_1^j, \mathbf{x}_{s_j+1}) \tag{6}$$

$$\vdots$$

$$\mathbf{h}_{d_j}^j = r(\mathbf{h}_{d_j-1}^j, \mathbf{x}_{n_j}) \tag{7}$$

where $\mathbf{h}_0$ denotes the initial hidden state, $d_j = n_j - s_j$ denotes the duration of the segment and $r(\cdot)$ is a non-linear function. We take the final hidden state $\mathbf{h}_{d_j}^j$ as the segment embedding vector, then $\Phi(\cdot)$ can be represented as

$$\Phi(y_j, \mathbf{e}_j, \mathbf{X}) = g(\mathbf{u}_j, \mathbf{h}_{d_j}^j), \tag{8}$$

where $g(\cdot)$ corresponds to one layer or multiple layers of linear or non-linear transformation. In fact, it is flexible to include other relevant features as additional inputs to the function $g(\cdot)$, e.g., the duration feature which can be obtained by converting d_j into another embedding vector. In practice, multiple RNN layers can be used transform the acoustic signal $\mathbf{X}$ before extracting the segment embedding vector $\mathbf{h}_{d_j}^j$ as Figure 1.

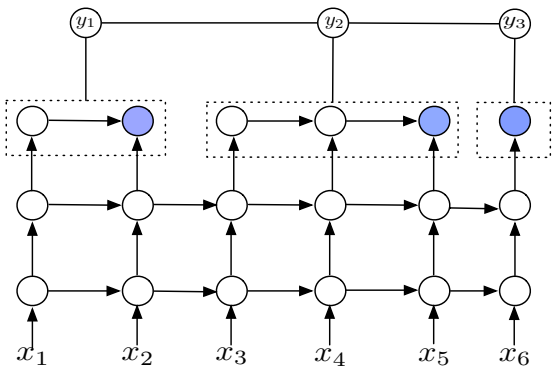

Figure 1: Segmental RNN using first-order CRF. The coloured circles denote the segment embedding vector $\mathbf{h}_{d_j}^j$ in Eq.(7). Using bi-directional RNNs is straightforward.

2.3. Conditional Maximum Likelihood Training

For speech recognition, the segmentation labels $\mathbf{E}$ are usually unknown, training the model by maximising the conditional probability as Eq. (1) is therefore not practical. The problem can be addressed by defining the loss function as the negative marginal log-likelihood as

$$\begin{aligned}
\mathcal{L}(\theta) &= -\log P(\mathbf{y} \mid \mathbf{X}) \\
&= -\log \sum_{\mathbf{E}} P(\mathbf{y}, \mathbf{E} \mid \mathbf{X}) \\
&= \underbrace{-\log \sum_{\mathbf{E}} \prod_{j} \exp f\left(y_j, \mathbf{e}_j, \mathbf{X}\right)}_{\equiv Z(\mathbf{X}, \mathbf{y})} + \log Z(\mathbf{X}),
\end{aligned} \tag{9}$$

where θ denotes the set of model parameters, and $Z(\mathbf{X}, \mathbf{y})$ denotes the summation over all the possible segmentations when only $\mathbf{y}$ is observed. To simplify notations, the objective function $\mathcal{L}(\theta)$ is define with only one training utterance.

However, the number of possible segmentations is exponential with the length of $\mathbf{X}$, which makes the naive computation of both $Z(\mathbf{X}, \mathbf{y})$ and $Z(\mathbf{X})$ impractical. Fortunately, this can be addressed by using the following dynamic programming algorithm as proposed in [15]:

$$\alpha_0 = 1 \tag{10}$$

$$\alpha_t = \sum_{0<k<t} \alpha_k \times \sum_{y \in \mathcal{Y}} f(y, \langle k, t \rangle, \mathbf{X}) \tag{11}$$

$$Z(\mathbf{X}) = \alpha_T \tag{12}$$

In Eq. (11), the first summation is over all the possible segmentation up to timestep t, and the second summation is over all the possible labels from the vocabulary. The computation cost of this algorithm is $O(T^2 \cdot |\mathcal{Y}|)$, where $|\mathcal{Y}|$ is the size of the vocabulary. The cost can be further reduced by introducing an upper bound of the segment length, in which case Eq. (11) can be rewritten as

$$\alpha_t = \sum_{l<k<t} \alpha_k \times \sum_{y \in \mathcal{Y}} f(y, \langle k, t \rangle, \mathbf{X}) \tag{13}$$

$$l = \begin{cases} 0 & \text{if} \quad t - L < 0 \\ t - L & \text{otherwise} \end{cases} \tag{14}$$

where L denotes the maximum value of the segment length. The cost is then reduced to $O(L \cdot T \cdot |\mathcal{Y}|)$, and for long sequences

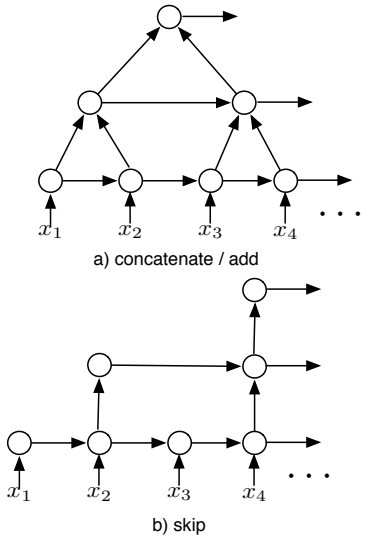

Figure 2: Hierarchical subsampling recurrent network [21] . The size of the subsampling window is two in this example.

like speech signals where $T \gg L$, the computational savings are substantial.

The term $Z(\mathbf{X}, \mathbf{y})$ can be computed similarly. In this case, since the label $\mathbf{y}$ is now observed, the summation over all the possible labels $y \in \mathcal{Y}$ in Eq. (11) is not necessary, i.e.,

$$\beta_{0,0} = 1 \tag{15}$$

$$\beta_{t,j} = \sum_{0<k<t} \beta_{k,j-1} \times f(y_j, \langle k, t \rangle, \mathbf{X}) \tag{16}$$

$$Z(\mathbf{X}, \mathbf{y}) = \beta_{T,J} \tag{17}$$

Again, we can limit the length of the possible segments as Eq. (13). Given $Z(\mathbf{X})$ and $Z(\mathbf{X}, \mathbf{y})$, the loss function $\mathcal{L}(\theta)$ can be minimised using the stochastic gradient decent (SGD) algorithm similar to training other neural network models. Other losses, for example, hinge, can be considered in future work.

2.4. Viterbi Decoding

During decoding, we need to search the target label sequence $\mathbf{y}$ that yields the highest posterior probability given $\mathbf{X}$ by marginalising out all the possible segmentations:

$$\mathbf{y}^* = \arg\max_{\mathbf{y}} \log \sum_{\mathbf{E}} P(\mathbf{y}, \mathbf{E} \mid \mathbf{X}) \tag{18}$$

This involves minor modification of the recursive algorithm in Eq. (11) that instead of summing over all the possible labels, the Viterbi path up to the timestep t is

$$\alpha_t^* = \sum_{0<k<t} \alpha_k^* \times \max_{y \in \mathcal{Y}} f(y, \langle k, t \rangle, \mathbf{X}) \tag{19}$$

However, marginalising out all the possible segmentations is still expensive. The computational cost can be further reduced by greedy searching the most likely segmentation, i.e.,

$$\alpha_t^* = \max_{0<k<t} \alpha_k^* \times \max_{y \in \mathcal{Y}} f(y, \langle k, t \rangle, \mathbf{X}), \tag{20}$$

which corresponds to the decoding objective as

$$\mathbf{y}^*, \mathbf{E}^* = \arg\max_{\mathbf{y}, \mathbf{E}} \log P(\mathbf{y}, \mathbf{E} \mid \mathbf{X}) \tag{21}$$

Table 1: Speedup by hierarchical subsampling networks.

subsampling	L	speedup
No	30	1
1 layer	15	$\sim$3x
2 layers	8	$\sim$10x

Table 2: Results of hierarchical subsampling networks. $d(\mathbf{w})$ and $d(\mathbf{h}_j)$ denote the dimension of $\mathbf{w}$ and $\mathbf{h}_{d_j}^j$ in Eqs. (3) and (7) respectively. `layers` denotes the number of LSTM layers and `hidden` is the dimension of the LSTM cells. `conc` is short for concatenating operation in the subsampling network.

System	$d(\mathbf{w})$	$d(\mathbf{h}_{d_j}^j)$	layers	hidden	PER(%)
`skip`	64	64	3	128	21.2
`conc`	64	64	3	128	21.3
`add`	64	64	3	128	23.2
`skip`	64	64	3	250	20.1
`conc`	64	64	3	250	20.5
`add`	64	64	3	250	21.5

This joint maximization algorithm may yield high search error, because it only considers one segmentation. In the future, we shall investigate the beam search algorithm which may yield a lower search error.

2.5. Further Speedup

It is computationally expensive for RNNs to model long sequences, and the number of possible segmentations is exponential with the length of the input sequence as mentioned before. The computational cost can be significantly reduced by using the hierarchical subsampling RNN [21] to shorten the input sequences, where the subsampling layer takes a window of hidden states from the lower layer as input as shown in Figure 2. In this work, we consider three variants: a) *concatenate* – the hidden states in the subsampling window are concatenated before been fed into the next layer; b) *add* – the hidden states are added into one vector for the next layer; c) *skip* – only the last hidden state in the window is kept and all the others are skipped. The last two schemes are computationally cheaper as they do not introduce extra model parameters.

3. Experiments

3.1. System Setup

We used the TIMIT dataset to evaluate the segmental RNN acoustic models. This dataset was preferred for the rapid evaluation of different system settings, and for the comparison to other CRF and end-to-end systems. We followed the standard protocol of the TIMIT dataset, and our experiments were based on the Kaldi recipe [22]. We used the core test set as our evaluation set, which has 192 utterances. We used 24 dimensional log fiterbanks (FBANKs) with delta and double-delta coefficients, yielding 72 dimensional feature vectors. Our models were trained with 48 phonemes, and their predictions were converted to 39 phonemes before scoring. The dimension of $\mathbf{u}_j$ was fixed to be 32. For all our experiments, we used the long short-term memory (LSTM) networks [23] as the implementation of RNNs, and the networks were always bi-directional. We set the initial SGD learning rate to be 0.1, and we exponentially decay the learning rate by a factor of 2 when the validation error stopped decreasing. Our models were trained with dropout

Table 3: Results of tuning the hyperparameters.

Dropout	$d(\mathbf{w})$	$d(\mathbf{h}^j_{d_j})$	layers	hidden	PER
0.2	64	64	3	128	21.2
	64	32	3	128	21.6
	32	32	3	128	21.4
	64	64	3	250	20.1
	64	32	3	250	20.4
	32	32	3	250	20.6
	64	64	6	250	19.3
	64	32	6	250	20.2
	32	32	6	250	20.2
0.1	64	64	3	128	21.3
	64	64	3	250	20.9
	64	64	6	250	20.4
×	64	64	6	250	21.9

Table 4: Results of three types of acoustic features.

Features	Deltas	$d(\mathbf{x}_t)$	PER
24-dim FBANK	√	72	19.3
40-dim FBANK	√	120	18.9
Kaldi	×	40	17.3

regularisation [24], using an specific implementation for recurrent networks [25]. The dropout rate was 0.2 unless specified otherwise. Our models were randomly initialised with the same random seed.

3.2. Results of Hierarchical Subsampling

We first demonstrate the results of the hierarchical subsampling recurrent network, which is the key to speed up our experiments. We set the size of the subsampling window to be 2, therefore each subsampling layer reduced the time resolution by a factor of 2. We set the maximum segment length L in Eq. (14) to be 300 milliseconds, which corresponded to 30 frames of FBANKs (sampled at the rate of 10 milliseconds). With two layers of subsampling recurrent networks, the time resolution was reduced by a factor of 4, and the value of L was reduced to be 8, yielding around 10 times speedup as shown in Table 1.

Table 2 compares the three implementations of the recurrent subsampling network detailed in section 2.5. We observed that concatenating all the hidden states in the subsampling window did not yield lower phone error rate (PER) than using the simple *skipping* approach, which may be due to the fact that the TIMIT dataset is small and it prefers a smaller model. On the other hand, adding the hidden states in the subsampling window together worked even worse, possibly due to that the sequential information in the subsampling window was flattened. In the following experiments, we sticked to the *skipping* method, and using two subsampling layers.

3.3. Hyperparameters and Different Features

We then evaluated the model by tuning the hyperparameters, and the results are given in Table 3. We tuned the number of LSTM layers, and the dimension of LSTM cells, as well as the dimensions of $\mathbf{w}$ and the segment vector $\mathbf{h}_j$. In general, larger models with dropout regularisation yielded higher recognition accuracy. Our best result was obtained using 6 layers of 250-dimensional LSTMs. However, without the dropout regularisation, the model can be easily overfit due to the small size of training set. In the future, we shall evaluate this model with a large dataset.

Table 5: Comparison to Related Works. `LM` denotes the language model, and `SD` denotes speaker-dependent transform. The HMM-DNN baseline was trained with cross-entropy using the Kaldi recipe. Sequence training did not improve it due to the small amount of data. Note that RNN transducer and attention-based RNN are equipped with built-in RNNLMs.

System	LM	SD	PER
HMM-DNN	√	√	18.5
first-pass SCRF [26]	√	×	33.1
Boundary-factored SCRF [27]	×	×	26.5
Deep Segmental NN [19]	√	×	21.9
Discriminative segmental cascade [28]	√	×	21.7
+ 2nd pass with various features	√	×	19.9
CTC [29]	×	×	18.4
RNN transducer [29]	-	×	17.7
Attention-based RNN [11]	-	×	17.6
Segmental RNN	×	×	18.9
Segmental RNN	×	√	17.3

We then evaluated another two types of features using the same system configuration that achieved the best result in Table 3. We increased the number of FBANKs from 24 to 40, which yielded slightly lower PER. We also evaluated the standard Kaldi features — 39 dimensional MFCCs spliced by a context window of 7, followed by LDA and MLLT transform and with feature-space speaker-dependent MLLR, which were the same features used in the HMM-DNN baseline in Table 5. The well-engineered features improved the accuracy of our system by more than 1% absolute.

3.4. Comparison to Related Works

In Table 5, we compare our result to other reported results using segmental CRFs as well as recent end-to-end systems. Previous state-of-the-art result using segmental CRFs on the TIMIT dataset is reported in [28], where the first-pass decoding was used to prune the search space, and the second-pass was used to re-score the hypothesis using various features including neural network features. Besides, the ground-truth segmentation was used in [28]. We achieved considerably lower PER with first-pass decoding, despite the fact that our CRF was zeroth-order, and we did not use any language model. Furthermore, our results are also comparable to that from the CTC and attention-based RNN end-to-end systems. The accuracy of segmental RNNs may be further improved by using higher-order CRFs or incorporating a language model into the decode step, and using beam search to reduce the search error.

4. Conclusions

In this paper, we present the segmental RNN — a novel acoustic model that combines the segmental CRF with an encoder RNN for end-to-end speech recognition. We discuss the practical training and decoding algorithms of this model for speech recognition, and the subsampling network to reduce the computational cost. Our experiments were performed on the TIMIT dataset, and we achieved strong recognition accuracy using zeroth-order CRF, and without using any language model. In the future, we shall investigate discriminative training criteria, and incorporating a language model into the decoding step. Future works also include implementing a weighted finite sate transducer (WFST) based decoder and scaling this model to large vocabulary datasets.

5. References

[1] D. Gillick, L. Gillick, and S. Wegmann, "Don't multiply lightly: Quantifying problems with the acoustic model assumptions in speech recognition," in *Proc. ASRU*. IEEE, 2011, pp. 71–76.

[2] M. Ostendorf, V. Digalakis, and O. Kimball, "From HMM's to segment models: A unified view of stochastic modeling for speech recognition," *IEEE Transactions on Speech and Audio Processing*, pp. 360–378, 1996.

[3] N. Smith and M. Gales, "Speech recognition using SVMs," in *Advances in neural information processing systems*, 2001, pp. 1197–1204.

[4] A. Gunawardana, M. Mahajan, A. Acero, and J. C. Platt, "Hidden conditional random fields for phone classification." in *INTERSPEECH*, 2005, pp. 1117–1120.

[5] Y. Hifny and S. Renals, "Speech recognition using augmented conditional random fields," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 17, no. 2, pp. 354–365, 2009.

[6] A. Graves and N. Jaitly, "Towards end-to-end speech recognition with recurrent neural networks," in *Proc. ICML*, 2014, pp. 1764–1772.

[7] A. Hannun, C. Case, J. Casper, B. Catanzaro, G. Diamos, E. Elsen, R. Prenger *et al.*, "Deep Speech: Scaling up end-to-end speech recognition," in *arXiv preprint arXiv:1412.5567*, 2014.

[8] H. Sak, A. Senior, K. Rao, and F. Beaufays, "Fast and accurate recurrent neural network acoustic models for speech recognition," in *Proc. INTERSPEECH*, 2015.

[9] Y. Miao, M. Gowayyed, and F. Metze, "EESEN: End-to-end speech recognition using deep RNN models and WFST-based decoding," in *Proc. ASRU*, 2015.

[10] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," in *Proc. ICLR*, 2015.

[11] J. K. Chorowski, D. Bahdanau, D. Serdyuk, K. Cho, and Y. Bengio, "Attention-based models for speech recognition," in *Advances in Neural Information Processing Systems*, 2015, pp. 577–585.

[12] L. Lu, X. Zhang, K. Cho, and S. Renals, "A study of the recurrent neural network encoder-decoder for large vocabulary speech recognition," in *Proc. INTERSPEECH*, 2015.

[13] W. Chan, N. Jaitly, Q. V. Le, and O. Vinyals, "Listen, attend and spell," *arXiv preprint arXiv:1508.01211*, 2015.

[14] L. Kong, C. Dyer, and N. A. Smith, "Segmental recurrent neural networks," *arXiv preprint arXiv:1511.06018*, 2015.

[15] S. Sarawagi and W. W. Cohen, "Semi-markov conditional random fields for information extraction," in *Advances in neural information processing systems*, 2004, pp. 1185–1192.

[16] J. Lafferty, A. McCallum, and F. Pereira, "Conditional random fields: Probabilistic models for segmenting and labeling sequence data," in *Proc. ICML*, 2001, pp. 282–289.

[17] G. Zweig, P. Nguyen, D. Van Compernolle, K. Demuynck, L. Atlas, P. Clark *et al.*, "Speech recognition with segmental conditional random fields: A summary of the JHU CLSP 2010 summer workshop," in *Proc. ICASSP*. IEEE, 2011, pp. 5044–5047.

[18] E. Fosler-Lussier, Y. He, P. Jyothi, and R. Prabhavalkar, "Conditional random fields in speech, audio, and language processing," *Proceedings of the IEEE*, vol. 101, no. 5, pp. 1054–1075, 2013.

[19] O. Abdel-Hamid, L. Deng, D. Yu, and H. Jiang, "Deep segmental neural networks for speech recognition." in *Proc. INTERSPEECH*, 2013, pp. 1849–1853.

[20] Y. He and E. Fosler-Lussier, "Segmental conditional random fields with deep neural networks as acoustic models for first-pass word recognition," in *Proc. INTERSPEECH*, 2015.

[21] A. Graves, "Hierarchical subsampling networks," in *Supervised Sequence Labelling with Recurrent Neural Networks*. Springer, 2012, pp. 109–131.

[22] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlıcek, Y. Qian, P. Schwarz, J. Silovský, G. Semmer, and K. Veselý, "The Kaldi speech recognition toolkit," in *Proc. ASRU*, 2011.

[23] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[24] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *The Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[25] W. Zaremba, I. Sutskever, and O. Vinyals, "Recurrent neural network regularization," *arXiv preprint arXiv:1409.2329*, 2014.

[26] G. Zweig, "Classification and recognition with direct segment models," in *Proc. ICASSP*. IEEE, 2012, pp. 4161–4164.

[27] Y. He and E. Fosler-Lussier, "Efficient segmental conditional random fields for phone recognition," in *Proc. INTERSPEECH*, 2012, pp. 1898–1901.

[28] H. Tang, W. Wang, K. Gimpel, and K. Livescu, "Discriminative segmental cascades for feature-rich phone recognition," in *Proc. ASRU*, 2015.

[29] A. Graves, A.-R. Mohamed, and G. Hinton, "Speech recognition with deep recurrent neural networks," in *Proc. ICASSP*. IEEE, 2013, pp. 6645–6649.