Segmental Encoder-Decoder Models for Large Vocabulary Automatic Speech Recognition

Eugen Beck, Mirko Hannemann, Patrick Doetsch, Ralf Schlüter, Hermann Ney

Human Language Technology and Pattern Recognition, Computer Science Department,
RWTH Aachen University, 52062 Aachen, Germany

{beck, mhannemann, doetsch, schlueter, ney}@cs.rwth-aachen.de

Abstract

It has been known for a long time that the classic Hidden-Markov-Model (HMM) derivation for speech recognition contains assumptions such as independence of observation vectors and weak duration modeling that are practical but unrealistic. When using the hybrid approach this is amplified by trying to fit a discriminative model into a generative one. Hidden Conditional Random Fields (CRFs) and segmental models (e.g. Semi-Markov CRFs / Segmental CRFs) have been proposed as an alternative, but for a long time have failed to get traction until recently. In this paper we explore different length modeling approaches for segmental models, their relation to attention-based systems. Furthermore we show experimental results on a handwriting recognition task and to the best of our knowledge the first reported results on the Switchboard 300h speech recognition corpus using this approach.

Index Terms: automatic speech recognition (ASR), (hidden) conditional random fields, segmental models, encoder-decoder, attention mechanism

1. Introduction

Classic Hidden-Markov-Models (HMMs) are known to be limited in the sense that the derivation of their decision rule assumes conditional independence of features given the state-sequence and only has weak duration modeling[1]. One way to alleviate these problems is to use segmental models. These generally assume a probability distribution that assigns a label w to a sequence of frames x in contrast to the framewise class conditional distribution $p(x|s)$ required by classic HMMs. Early works on this topic include [2] and [3]. Most recent works are based on [4], where Semi-Markov CRFs are presented. No explicit length modeling is done in this approach. The feature functions used in the model may include length features, but the model is normalized over the set of output labels.

In [5] a set of hand-engineered segment features is used to improve upon a baseline HMM system on data from Bing Mobile voice-search. The authors of [6] also use this framework but model the joint probability distribution of labels and segmentation. The authors introduce boundary features that result in a reduction in training and evaluation complexity. The model is evaluated on the TIMIT corpus, in contrast to the previously mentioned approaches which use the model for re-scoring, the model is applied in the first-pass decoding. In [7] and [8] the authors focus on evaluating various deep neural network (DNN) architectures to model the joint probability distribution of segment boundaries and labels. Variability in segment length is handled by evaluating a fixed number of DNNs evenly spread across the segment. In [9] a multi-pass approach to incorporate higher-order features is used. The publications [10] and [11] present work using an architecture similar to ours (and similar also [12, 13]), introducing a stochastic alignment. A series of bidirectional long short-term memory (LSTM) layers is followed by a decoder layer. The hidden state of this decoder layer is used in the feature-function to determine the joint probability of label and segmentation. In [10] a handwriting task [14] and a Chinese-character segmentation task and Part-Of-Speech-tagging are used to evaluate the model. [11] evaluate their work on the TIMIT corpus. In [15] higher order features are integrated by changing the decoder. It does not see every frame of a segment, only a fixed number of frames (similar to [7, 8]). This allows it to iterate over multiple segments, creating a higher-order CRF. Evaluation (given segment boundaries) is again on TIMIT. In [16] a different aspect of segment modeling is addressed. In order to improve training convergence and final performance multi-task learning is applied by simultaneously optimizing a CTC-loss [17]. Again TIMIT is used for evaluation. In previous work from our group [18, 19] inverted alignments were introduced as a form of deep Segmental CRFs. For these studies the length model was static and did not incorporate information from the input features.

In this paper we address the problem of integrating discriminative neural network based acoustic models properly into a speech recognition system by using a segmental approach. In the derivation of this approach the need to model segment-lengths occurs naturally and we investigate various ways to realize expressive length-models. Evaluations are first performed on a handwriting task and the two most successful models are then evaluated on the Switchboard English 300h task. These are, to the best of our knowledge, the first results of a Segmental CRF system on this task. In addition we compare this approach to the recently introduced attention mechanism from a modeling perspective.

The rest of the paper is structured as follows: First we derive different decision rules for classic HMMs, Segmental CRFs and recently introduced attention-systems and point out the major differences and similarities. Next we describe the neural network architecture and its training. Afterwards we show experimental results for a handwriting task and Switchboard English 300h. An outlook into future work and a conclusion finishes the paper.

2. Model derivation

The classic Bayes decision rule r for sequence-to-sequence classification is:

$$r : x_1^T \rightarrow w_1^N = \arg\max_{w_1^N} p(w_1^N | x_1^T)$$

where x_1^T is a sequence of observation vectors of length T and w_1^N a sequence of output labels of length N. From this starting point we look at three different modeling approaches: Hybrid-HMM, segmental models and attention models.

2.1. Hybrid-HMM

In classic HMM-Systems [20], a sequence of states s_1^T is introduced as hidden variables and we assume conditional independence of observations given the state[1]. To use a discriminative acoustic model we need to use Bayes identity to swap the dependence between state and observation.

$$\begin{aligned}
\arg\max_{w_1^N} p(w_1^N|x_1^T) &= \arg\max_{w_1^N} \left\{ p(w_1^N) \cdot p(x_1^T|w_1^N) \right\} \\
p(w_1^N) \cdot p(x_1^T|w_1^N) &= p(w_1^N) \cdot \sum_{s_1^T} p(s_1^T, x_1^T|w_1^N) \\
&= p(w_1^N) \cdot \sum_{s_1^T} \prod_t p(x_t|s_t) \cdot p(s_t|s_{t-1}) \\
&= p(w_1^N) \cdot \sum_{s_1^T} \prod_t \frac{p(s_t|x_t)}{p(s_t)} \cdot p(s_t|s_{t-1}) \\
&= p(w_1^N) \cdot \max_{s_1^T} \prod_t \frac{p(s_t|x_t)}{p(s_t)} \cdot p(s_t|s_{t-1})
\end{aligned}$$

2.2. Segmental Model

Segmental models are sometimes derived by defining feature functions on sub-sequences of x_1^T (e.g. [5]) that are used in a log-linear model. Alternatively, we can view the segmentation as the introduction of a hidden variable of label boundaries t_0^N such that output label w_n is assigned the features $x_{t_{n-1}+1}^{t_n}$. We assume $t_0 = 0, t_N = T$ (coverage) and $t_n < t_m \Leftrightarrow n < m$ (monotonicity). This is analogous to [18].

$$\begin{aligned}
\arg\max_{w_1^N} p(w_1^N|x_1^T) &= \arg\max_{w_1^N} \left\{ \sum_{t_0^N} p(w_1^N, t_0^N|x_1^T) \right\} \\
&= \arg\max_{w_1^N} \left\{ \sum_{t_0^N} \prod_n p(w_n, t_n|w_1^{n-1}, t_0^{n-1}, x_1^T) \right\} \\
&\quad \text{(first order Markov assumption)} \\
&= \arg\max_{w_1^N} \left\{ \sum_{t_0^N} \prod_n p(w_n, t_n|w_{n-1}, t_{n-1}, x_1^T) \right\} \\
&\quad \text{(maximum approximation for segmentation)} \\
&= \arg\max_{w_1^N, t_0^N} \left\{ \prod_n p(w_n, t_n|w_{n-1}, t_{n-1}, x_1^T) \right\}
\end{aligned}$$

In practice we do not estimate absolute positions t_n, but instead estimate $\Delta t_n = t_n - t_{n-1}$. If the input for the estimator is dependent on t_n and t_{n-1} we do not write Δt_n but stick to t_n.

From this base form we can derive multiple variants:

$p(w_n, t_n|w_{n-1}, t_{n-1}, x_1^T) = \cdots$

1. Static length model:
   $\cdots = p(w_n|t_n, t_{n-1}, x_1^T) \cdot p(\Delta t_n)$
2. Data dependent length model:
   $\cdots = p(w_n|t_n, t_{n-1}, x_1^T) \cdot p(t_n|t_{n-1}, x_1^T)$
3. Joint distribution without label feedback:
   $\cdots = p(w_n, t_n|t_{n-1}, x_1^T)$
4. Label dependent length model:
   $\cdots = p(w_n|t_{n-1}, x_1^T) \cdot p(t_n|w_n, t_{n-1}, x_1^T)$

For the first three variants we limit the maximum length of each segment to improve computational performance. In the fourth variant, similar to a geometric distribution, we can model $p(t_n|w_n, t_{n-1}, x_1^T)$ using a frame-by-frame decision (continue or terminate). This results in a framewise criterion and thus an enforcement of a maximum segment length is not necessary. The label-dependent framewise termination probability $p(end|t, w_n, x_1^T)$ is realized with one sigmoid unit per label.

$$p(t_n|w_n, t_{n-1}, x_1^T) = \left(\prod_{t=t_{n-1}+1}^{t_{n-1}} 1 - p(end|t, w_n, x_1^T) \right) \cdot p(end|t_n, w_n, x_1^T)$$

We would like to point out that in tasks like speech or handwriting recognition it is useful to keep the language model $p(w_1^N)$ on the level of output symbols (words), but use smaller units (characters, mono-/(tied) triphones) for the acoustic model. Thus, the language model can be trained on larger text corpus and during decoding, the acoustic model scores and language model scores are combined in a log-linear fashion (mixture of experts).

All of the previous derivations where done with a fixed N. However, during decoding the sentence length is unknown and thus the full decision rule is:

$$\begin{aligned}
r : x_1^T \rightarrow w_1^N &= \arg\max_{N, w_1^N} p(N, w_1^N|x_1^T) \\
&= \arg\max_{N, w_1^N} p(N|x_1^T) \cdot p(w_1^N|N, x_1^T)
\end{aligned}$$

If no sentence-length model is used, short sequences with low individual probability per label might have a higher total probability than long sequences with high probability per label because long sequences contain more factors. This is a known problem and is sometimes fixed by adding a length penalty e.g. in [21].

2.3. Attention Mechanism

Acoustic models using what is called an *attention mechanism* have recently been proposed [22, 23, 24, 25, 21, 26, 27] and have recently been able to outperform state-of-the art hybrid HMM systems[28].

The decision rule is as follows:

$$\begin{aligned}
\arg\max_{w_1^N} p(w_1^N|x_1^T) &= \prod_{n=1}^{N} p(w_n|w_1^{n-1}, x_1^T) \\
&= \prod_{n=1}^{N} p(w_n|w_1^{n-1}, c_n(w_1^{n-1}, x_1^T))
\end{aligned}$$

Where c_n is the output of the attention mechanism which uses attention weights $\alpha_{t,n}$ derived from an energy function $e_{t,n}(w_1^n, x_1^T)$ and the encoder output $h_t(x_1^T)$ at time t.

Similar to variants 1-3 of our model, some attentions systems limit the attention weights to a window of fixed length

[1] For consistency with our previous publications, in the following equations, we denote model assumptions with the equal sign.

[24, 27]. Other use a gating mechanism to limit the attention [22]. The approach that is most similar to ours is hard monotonic attention [26]. There the authors use a stochastic process to determine the window size based on a framewise sampling and then focus the attention weights exclusively on the last frame within this window. The framewise decision to extend the window is similar to what we do in variant 4 and similar to the emit/shift decision in [12], except that we evaluate the label probability at the beginning of the window and include the length probability in the probability of the sequence.

3. Neural Network Architecture and Training

We build our neural networks similar to the ones described in the above mentioned publications. Several layers of bidirectional LSTM layers[29] form the encoder, which transforms the inputs sequence x_1^T into a sequence h_1^T. Optionally, downsampling is performed between the first and second layer by average pooling two consecutive frames, resulting in an output sequence $h_1^{[T/2]}$. Based on this encoder output we use a single layer forward LSTM-layer to estimate various probability distributions in the segmental model. Distributions over length use a softmax that spans multiple time-frames.

Variant 4 of our model has two components: the label prediction posterior $p(w_n|t_{n-1}, x_1^T)$, which is trained with targets at the beginning of a segment, and the framewise termination probability $p(end|t, w_n, x_1^T)$, which is realized by a sigmoid unit for every label, trained using framewise targets by maximizing the logarithm of the corresponding sigmoid units at those positions. The label prediction posterior and termination probability were optimized in a multi-task fashion by connecting both output layers to the same encoder network.

Training can be performed by maximizing the cross-entropy on sequence level:

$$\mathcal{L} = \sum_r \log p\left(\left(w_1^N\right)_r \mid \left(x_1^T\right)_r\right)$$

This can be done by expectation-maximization, i.e. optimizing the ratio of two parameter sets p and q.

$$\begin{aligned}
\log \frac{p(w_1^N|x_1^T)}{q(w_1^N|x_1^T)} &= \sum_{t_0^N} q(t_0^N|w_1^N, x_1^T) \log \frac{p(w_1^N|x_1^T)}{q(w_1^N|x_1^T)} \\
&\geq \sum_{t_0^N} q(t_0^N|w_1^N, x_1^T) \log \frac{p(w_1^N|x_1^T)}{q(w_1^N|x_1^T)} \\
&\quad - \underbrace{\sum_{t_0^N} q(t_0^N|w_1^N, x_1^T) \log \frac{q(t_0^N|w_1^N, x_1^T)}{p(t_0^N|w_1^N, x_1^T)}}_{\geq 0 \text{ (KL-divergence)}} \\
&= \sum_{t_0^N} q(t_0^N|w_1^N, x_1^T) \log \frac{p(w_1^N, t_0^N|x_1^T)}{q(w_1^N, t_0^N|x_1^T)} \\
&= Q(p,q) - Q(q,q)
\end{aligned}$$

With $Q(p,q) = \sum_{t_0^N} q(t_0^N|w_1^N, x_1^T) \log p(w_1^N, t_0^N|x_1^T)$.

When we insert our model assumptions we get:

$$\begin{aligned}
Q(p,q) &= \sum_{t_0^N} q(t_0^N|w_1^N, x_1^T) \log \prod_{n=1}^{N} p(w_n, t_n|t_{n-1}, x_1^T) \\
&= \sum_{n,t,t'} \sum_{\substack{t_0^N: t_n=t, \\ t_{n-1}=t'}} q(t_0^N|w_1^N, x_1^T) \log p(w_1^N, t_0^N|x_1^T) \\
&= \sum_{n,t,t'} \gamma_n(t, t'|w_1^N, x_1^T) \log p(w_1^N, t_0^N|x_1^T)
\end{aligned}$$

Then we get the following derivative for e.g. the joint probability distribution model:

$$\frac{\partial Q(p,q)}{\partial p(w, t|t', x_1^T)} = \frac{1}{p(w, t|t', x_1^T)} \sum_{n: w_n = w} \gamma_n(t, t'|w_1^N, x_1^T)$$

γ_n can be computed via the well known forward-backward algorithm, or if we apply the Viterbi-approximation to the sum over t_0^N we arrive at a single time segmentation. Some of our early experiments showed that this approximation can result in the NN displaying a high confidence for arbitrary labels in segments that were not seen during training. This problem did not occur while training the framewise model (Variant 4). As training from scratch using the Baum-Welch algorithm did not converge reliably we used an alignment generated by another system as a starting point to approximate γ_n. Given a framewise alignment α_1^L we define

$$\psi(w, t|t') = \begin{cases} 1 \text{ if } t = \arg\min_{\tau > t'} \alpha_\tau \neq \alpha_{\tau+1} \wedge w = \alpha_{t'} \\ 0 \text{ else} \end{cases}$$

We train the model by optimizing

$$\mathcal{L} = \sum_{t,t',c} \psi(w, t|t') \log p(w, t|t', x_1^T)$$

4. Experiments

The search for the best word sequence for our segmental models is done using with several modified versions of the dynamic tree-search decoder in [30]. More details on an older version of our inverted decoder can be found in [19].

4.1. IAM

We first present results on the IAM Handwriting database [31], which is based on the Lancaster-Oslo/Bergen (LOB) corpus. It consists of a train, validation and evaluation part, which contain 6482, 976 and 2915 lines of text, respectively. We use a 50k lexicon with an OOV-rate of 4.01% / 3.47% on the validation / evaluation part, respectively. No handling of OOVs (i.e. using sub-words) has been done in our model, thus they are responsible for the majority of the errors. The language model is a 5-gram count-based model. As input to our model, we used frames extracted from a sliding window of 8×32 pixels and a shift of 3. We then applied a moment-based size normalization together with a dimensionality reduction to 20 principal components. The acoustic (visual) model was trained on an alignment generated by a previously trained LSTM hybrid system.

The encoder used in this task is a 4-layer BLSTM with 512 units per direction. The decoder for variants 1-3 is a 256-cell mono-directional LSTM layer. We tried (and optimized) different length models (Table 1). As could be expected a static

Model	valid-set	eval-set
baseline (hybrid, vsc)	10.9%	14.4%
Variant 1 (single state per character)	15.4%	20.5%
Variant 2 (single state per character)	13.5%	17.5%
Variant 3 (single state per character)	12.6%	16.6%
Variant 3 (vsc + length-norm.)	11.7%	14.9%
Variant 4 (single state per character)	12.5%	16.1%

Table 1: *Results on the IAM database, all results are WER, vsc = variable number of states per character*

length model (Variant 1) performed worst. It was estimated on the alignment of the training corpus. A separate decoding layer (shared encoder, multi-task learning) for length modeling reduced the WER by 1.9% absolute (Variant 2). Next we removed the length model decoder again and changed the softmax for the label decoder to normalize over all durations, thus estimating the joint distribution of labels and lengths (Variant 3). This improved performance by 0.9% absolute. The framewise model (Variant 4) achieved basically the same performance. Until now we used a single state per character. Our baseline uses a varying number of states per character (VSC) (depending on the average width of the symbol). The resulting average number of states per symbol is 7.13. This model performed better when we added length normalization (multiplying by segment length in log-domain). This is due to the fact that these sub-character segments tend to be very short and our window size can easily span many of these sub-characters. In this setting it can occur that a single "bad" hypothesis can receive a better score than multiple hypothesis that might locally have the highest probability. With length normalization and VSC we get another 0.9% absolute improvement. This system is only 0.8%/0.5% worse than our very competitive baseline.

Variant 4 using models with 1-state per character is a model that requires little tuning - the most important parameters are the language model scale and a calibration exponent. The language model scale for these models is usually around 1.0-2.0, as a rule-of-thumb it corresponds to the language model scale of the hybrid baseline dividend by the average character length. A calibration exponent β is applied to the framewise termination probability (usually around 0.5) and effectively balances insertions and deletions on a character level:

$$p'(end|t, w_n, x_1^T) = p(end|t, w_n, x_1^T)^\beta$$
$$p'(\overline{end}|t, w_n, x_1^T) = 1 - p'(end|t, w_n, x_1^T)$$

4.2. Switchboard

The NN architecture was the same as the one used for the IAM experiments, except for network depth. More details on the BLSTM training can be found in [32]. We train our models on Gammatone features [33] extracted from the 300h Switchboard-1 Release 2 and evaluate performance on the Hub5'00 corpus. The language model is a 4-gram count-based model trained on the transcripts of the training and Fisher English corpora. The alignment used for training was generated by a sequence-discriminatively trained tandem system. Our best baseline system uses a classification and regression tree (CART) to tie 3-state triphones into a fixed number of acoustic models. For our hybrid setup we use a CART with 9001 clusters. As the joint model (Variant 3) performs a softmax over the the segment length and the labels, a large CART is too costly during training. Thus we also estimated a smaller CART (with only 1501 clusters) and tested monophones. As the encoder can see past segment boundaries the NN could be able to incorporate knowledge of past and future acoustic events into the decision making process.

Model	target	DS	LN	CH	SWB	Σ
hybrid	1-sta. monophone	no	N/A	27.3	14.0	20.6
	1-sta. 1501-CART	no	N/A	23.6	12.1	17.8
	3-sta. 9001-CART	no	N/A	21.6	11.0	16.3
Var. 3	1-sta. monophones	no	no	42.5	22.1	32.3
		yes	yes	34.5	17.6	26.1
	3-sta. monophone	no	yes	34.9	19.3	27.4
	1-sta. 1501-CART	yes	yes	29.6	15.0	22.3
Var. 4	3-sta. 9001-CART	no	no	25.6	13.0	19.3

Table 2: *Results on the Hub5'00 evaluation corpus in terms of %WER, DS=downsampling, LN=length normalization, CH=Callhome part, SWB=Switchboard part, sta. = state*

For the Switchboard experiments, additionally to the language model scale as tuning parameter, we introduced a silence penalty to balance deletions and insertions, and we limited the maximum segment length to values around 12-25, depending on downsampling and number of states. For the Var. 3 1-state monophone model we experimented with extra penalties for long silence segments and got some small improvements.

In Table 2 we see results for The Hub5'00 evaluation corpus. As expected our baseline system performs better when output labels are more fine-grained. A similar trend can be observed for segment-based models. The newly presented models do not yet reach the performance of our baseline system, so more work has to be done to examine possible improvements. One limiting factor for us was training time. The training time increases proportionally to the maximum window size (for Variant 1-3).

5. Future Work

The systems we presented in this paper were all trained given a fixed alignment. In future we want to investigate ways to train our models end-to-end from scratch. We have found that modeling the length of a sequence seems to be important in some settings. Thus we will investigate ways to incorporate this into our decoder, probably by decoding in a label-synchronous fashion.

6. Conclusions

In this paper we have presented a framework to embed discriminative acoustic models into a mathematically sound framework for ASR. We examined various length models within this framework and while the results on Switchboard are not state-of-the-art yet, we believe that they do warrant further research into this topic. The connection to attention systems is also intriguing as they represent a similar approach.

7. Acknowledgments

The research was partially supported by a Google PhD fellowship grant and has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement No 694537) and under the Marie Skłodowska-Curie grant agreement No 644283. The work reflects only the authors' views and neither the European Research Council Executive Agency nor Google is not responsible for any use that may be made of the information it contains.

8. References

[1] M. Ostendorf, V. V. Digalakis, and O. A. Kimball, "From HMM's to segment models: a unified view of stochastic modeling for speech recognition," *IEEE Transactions on Speech and Audio Processing*, vol. 4, no. 5, pp. 360–378, Sep 1996.

[2] M. Bush and G. Kopec, "Network-based connected digit recognition," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 35, no. 10, pp. 1401–1413, Oct 1987.

[3] M. Russell, "A segmental HMM for speech pattern modelling," in *1993 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, vol. 2, April 1993, pp. 499–502 vol.2.

[4] S. Sarawagi and W. W. Cohen, "Semi-markov conditional random fields for information extraction," in *Advances in Neural Information Processing Systems 17*, L. K. Saul, Y. Weiss, and L. Bottou, Eds. MIT Press, 2005, pp. 1185–1192. [Online]. Available: http://papers.nips.cc/paper/2648-semi-markov-conditional-random-fields-for-information-extraction.pdf

[5] G. Zweig and P. Nguyen, "A segmental CRF approach to large vocabulary continuous speech recognition," in *2009 IEEE Workshop on Automatic Speech Recognition Understanding*, Nov 2009, pp. 152–157.

[6] Y. He and E. Fosler-Lussier, "Efficient segmental conditional random fields for one-pass phone recognition," in *Interspeech 2012*, 2012, pp. 1898–1901.

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

[8] Y. He and E. Fosler-Lussier, "Segmental conditional random fields with deep neural networks as acoustic models for first-pass word recognition," in *Interspeech 2015*, 2015, pp. 2640–2644.

[9] H. Tang, W. Wang, K. Gimpel, and K. Livescu, "Discriminative segmental cascades for feature-rich phone recognition," in *2015 IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, Dec 2015, pp. 561–568.

[10] L. Kong, C. Dyer, and N. A. Smith, "Segmental recurrent neural networks," in *Intern. Conf. on Learning Representations (ICLR)*, 2016.

[11] L. Lu, L. Kong, C. Dyer, N. A. Smith, and S. Renals, "Segmental recurrent neural networks for end-to-end speech recognition," in *Interspeech 2016*, 2016, pp. 385–389. [Online]. Available: http://dx.doi.org/10.21437/Interspeech.2016-40

[12] L. Yu, J. Buys, and P. Blunsom, "Online segment to segment neural transduction," in *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2016.

[13] L. Yu, P. Blunsom, C. Dyer, E. Grefenstette, and T. Kocisky, "The neural noisy channel," in *Intern. Conf. on Learning Representations (ICLR)*, 2017.

[14] R. H. Kassel, "A comparison of approaches to on-line handwritten character recognition," Ph.D. dissertation, Massachusetts Institute of Technology, Cambridge, MA, USA, 1995, not available from University Microfilms Int.

[15] M. Ratajczak, S. Tschiatschek, and F. Pernkopf, "Frame and segment level recurrent neural networks for phone classification," in *Proc. Interspeech 2017*, 2017, pp. 1318–1322. [Online]. Available: http://dx.doi.org/10.21437/Interspeech.2017-1064

[16] L. Lu, L. Kong, C. Dyer, and N. A. Smith, "Multitask Learning with CTC and Segmental CRF for Speech Recognition," in *Interspeech 2017*, 2017, pp. 954–958. [Online]. Available: http://dx.doi.org/10.21437/Interspeech.2017-71

[17] A. Graves, S. Fernández, F. Gomez, and J. Schmidhuber, "Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural networks," in *Proceedings of the 23rd International Conference on Machine Learning*, ser. ICML '06. New York, NY, USA: ACM, 2006, pp. 369–376. [Online]. Available: http://doi.acm.org/10.1145/1143844.1143891

[18] P. Doetsch, S. Hegselmann, R. Schlüter, and H. Ney, "Inverted HMM - a Proof of Concept," in *Neural Information Processing Systems Workshop*, Barcelona, Spain, Dec 2016.

[19] P. Doetsch, M. Hannemann, R. Schlüter, and H. Ney, "Inverted alignments for end-to-end automatic speech recognition," *IEEE Journal of Selected Topics in Signal Processing*, vol. 11, no. 8, pp. 1265–1273, Dec 2017.

[20] F. Jelinek, *Statistical methods for speech recognition*. MIT press, 1997.

[21] W. Chan, N. Jaitly, Q. Le, and O. Vinyals, "Listen, attend and spell: A neural network for large vocabulary conversational speech recognition," in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, March 2016, pp. 4960–4964.

[22] J. Chorowski, D.Bahdanau, K. Cho, and Y. Bengio, "End-to-end continuous speech recognition using attention-based recurrent nn: First results," in *Deep Learning and Representation Learning Workshop: NIPS 2014*, Montreal, Canada, Dec 2014.

[23] Y. Miao, M. Gowayyed, and F. Metze, "EESEN: End-to-end speech recognition using deep RNN models and WFST-based decoding," in *2015 IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, Dec 2015, pp. 167–174.

[24] 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 28*, C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, Eds. Curran Associates, Inc., 2015, pp. 577–585. [Online]. Available: http://papers.nips.cc/paper/5847-attention-based-models-for-speech-recognition.pdf

[25] D. Bahdanau, J. Chorowski, D. Serdyuk, P. Brakel, and Y. Bengio, "End-to-end attention-based large vocabulary speech recognition," in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, March 2016, pp. 4945–4949.

[26] C. Raffel, M.-T. Luong, P. J. Liu, R. J. Weiss, and D. Eck, "Online and linear-time attention by enforcing monotonic alignments," in *Proceedings of the 34th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, D. Precup and Y. W. Teh, Eds., vol. 70. International Convention Centre, Sydney, Australia: PMLR, 06–11 Aug 2017, pp. 2837–2846. [Online]. Available: http://proceedings.mlr.press/v70/raffel17a.html

[27] C.-C. Chiu and C. Raffel, "Monotonic chunkwise attention," in *International Conference on Learning Representations*, 2018. [Online]. Available: https://openreview.net/forum?id=Hko85plCW

[28] C.-C. Chiu, T. N. Sainath, Y. Wu, R. Prabhavalkar, P. Nguyen, Z. Chen, A. Kannan, R. J. Weiss, K. Rao, E. Gonina, N. Jaitly, B. Li, J. Chorowski, and M. Bacchiani, "State-of-the-art Speech Recognition With Sequence-to-Sequence Models," *ArXiv e-prints*, dec 2017. [Online]. Available: https://arxiv.org/abs/1712.01769

[29] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," *Neural computation,* Vol. 9, No. 8, pp. 1735–1780, 1997.

[30] D. Rybach, S. Hahn, P. Lehnen, D. Nolden, M. Sundermeyer, Z. Tüske, S. Wiesler, R. Schlüter, and H. Ney, "RASR - The RWTH Aachen University Open Source Speech Recognition Toolkit," in *IEEE Automatic Speech Recognition and Understanding Workshop*, Waikoloa, HI, USA, Dec 2011.

[31] U.-V. Marti and H. Bunke, "The IAM-database: an English sentence database for offline handwriting recognition," *International Journal on Document Analysis and Recognition*, vol. 5, no. 1, pp. 39–46, Nov 2002. [Online]. Available: https://doi.org/10.1007/s100320200071

[32] A. Zeyer, P. Doetsch, P. Voigtlaender, R. Schlüter, and H. Ney, "A comprehensive study of deep bidirectional LSTM RNNs for acoustic modeling in speech recognition," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2017, pp. 2462–2466.

[33] R. Schlüter, I. Bezrukov, H. Wagner, and H. Ney, "Gammatone features and feature combination for large vocabulary speech recognition," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, vol. 4, April 2007, pp. IV–649–IV–652.