MONOTONIC SEGMENTAL ATTENTION FOR AUTOMATIC SPEECH RECOGNITION

Albert Zeyer\textsuperscript{1,2}, Robin Schmitt\textsuperscript{1,2}, Wei Zhou\textsuperscript{1,2}, Ralf Schlüter\textsuperscript{1,2}, Hermann Ney\textsuperscript{1,2}

\textsuperscript{1}Human Language Technology and Pattern Recognition, Computer Science Department, RWTH Aachen University, 52062 Aachen, Germany, \textsuperscript{2}AppTek GmbH, 52062 Aachen, Germany

\{zeyer,zhou,schlueter,ney\}@cs.rwth-aachen.de, robin.schmitt1@rwth-aachen.de

ABSTRACT

We introduce a novel segmental-attention model for automatic speech recognition. We restrict the decoder attention to segments to avoid quadratic runtime of global attention, better generalize to long sequences, and eventually enable streaming. We directly compare global-attention and different segmental-attention modeling variants. We develop and compare two separate time-synchronous decoders, one specifically taking the segmental nature into account, yielding further improvements. Using time-synchronous decoding for segmental models is novel and a step towards streaming applications. Our experiments show the importance of a length model to predict the segment boundaries. The final best segmental-attention model using segmental decoding performs better than global-attention, in contrast to other monotonic attention approaches in the literature. Further, we observe that the segmental model generalizes much better to long sequences of up to several minutes.

Index Terms—Segmental attention, segmental models

1. INTRODUCTION & RELATED WORK

The attention-based encoder-decoder architecture \cite{bahdanau2014neural} has been very successful as an end-to-end model for many tasks including speech recognition \cite{graves2013speech, graves2014speech, yang2015attention, zeyer2018monotonic}. However, for every output label, the attention weights are over all the input frames, referred to as global attention. This has the drawbacks of quadratic runtime-complexity, potential non-monotonicity; it does not allow for online streaming recognition and it does not generalize to longer sequence lengths than those seen in training \cite{srinivasan2017streaming, earle2019streaming, xu2015streaming}.

There are many attempts to solve some of these issues. Monotonic chunkwise attention (MoChA) \cite{richard2017segmental, richard2018mocha, richard2019segmental} is one popular approach which uses fixed-size chunks for soft attention and a deterministic approach for the chunk positions, i.e., the position is not treated as a latent variable in recognition. Many similar approaches using a local fixed-sized window and some heuristic or separate neural network for the position prediction were proposed \cite{richard2017segmental, richard2018mocha, richard2019segmental, dinan2018efficient, kim2019so, li2020joint, diakogiannis2020hybrid, dinan2020non}. The attention sometimes also uses a Gaussian distribution which allows for a differentiable position prediction \cite{hannun2018deep, k方面的中文翻译问题，可能需要调整句子结构以使其更自然。}

Some models add a penalty in training, or are extended to have an implicit bias to encourage monotonicity \cite{richard2017segmental, richard2018mocha, richard2019segmental}. Framewise defined models like CTC \cite{graves2006connectionist} or transducers \cite{graves2005sequence} canonically allow for streaming, and there are approaches to combine such framewise model with attention \cite{stafylakis2015speech, srinivasan2017streaming, earle2019streaming, xu2015streaming}.

Our probabilistic formulation using latent variables for the segment boundaries is similar to other segmental models \cite{graves2016hybrid, visweswara2017segmental, zeyer2019streaming, zeyer2019segmental}, although attention has not been used on the segments except in \cite{zeyer2019streaming} and there are usually more independence assumptions such as first or zero order dependency on the label, and only a first order dependency on the segment boundary, which is a critical difference. It has also been shown that transducer models and segmental models are equivalent \cite{zeyer2019streaming}.

Here, we want to make use of the attention mechanism while solving the mentioned global attention drawbacks by making the attention local and monotonic on segments. We treat the segment boundaries as latent variables and end up at our segmental attention models. Such segmental models by definition are monotonic, allow for streaming, and are much more efficient by using only local attention. Our aim is to get a better understanding of such segmental attention models, to directly compare it to the global attention model, to study how to treat and model the latent variable, how to perform the search for recognition, how to treat silence, how well it generalizes to longer sequences among other questions.

2. GLOBAL ATTENTION MODEL

Our starting point is the standard global-attention-based encoder-decoder model \cite{bahdanau2014neural} adapted for speech recognition \cite{graves2013speech, graves2014speech, yang2015attention, zeyer2018monotonic}, specifically the model as described in \cite{graves2016hybrid, yu2018global}. We use an LSTM-based \cite{hochreiter1997long} encoder which gets a sequence of audio feature frames $x^T_1$ of length $T'$ as input and encodes it as a sequence

$$h^T_1 = \text{Encoder}(x^T_1)$$

of length $T$, where we apply downsampling by max-pooling in time inside the encoder by factor 6. For the output label sequence $a^S_1$ of length $S$, given the encoder output sequence $h^T_1$ of length $T$, we define

$$p(a^S_1 | h^T_1) = \prod_{s=1}^{S} p(a_s | a^{s-1}_s, h^T_1),$$

where $p(a_s | a^{s-1}_s, h^T_1)$ is the label model.
The label model uses global attention on $h^T_t$ per each output step $s$. The neural structure which defines the probability distribution of the labels is also called the decoder. The decoder of the global-attention model is almost the same as our segmental attention model, which we will define below in detail. The main difference is segmental vs. global attention.

### 3. OUR SEGMENTAL ATTENTION MODEL

Now, we introduce *segment boundaries* as a sequence of latent variables $t^S_t$. Specifically, for one output $a_s$, the segment is defined by $[t_s-1+1, t_s]$, with $t_0 = 0$ and $t_T = T$ fixed, and we require $t_s > t_{s-1}$ for strict monotonicity. Thus, the segmentation fully covers all frames of the sequence. One such segment is highlighted in Figure 1. The model now uses attention only in that segment, i.e. on $h^T_{t_{s-1}+1}$. For the output label sequence $a^S_t$, we now define the segmental model as

$$p(a^S_t \mid h^T_T) = \prod_{s=1}^{S} p(t_s \mid \ldots, \alpha^S_{a_s} \mid a^S_{t_{s-1} + 1}, h^T_{t_{s-1} + 1}, \ldots).$$

In the simplest case, we even do not use any length model ($\alpha = 0$). The intuition was that a proper dynamic search over the segment boundaries can be guided by the label model alone, as it should produce bad scores for bad segment boundaries. We also test a simple static length model, and a neural length model, as we will describe later. The label model is mostly the same as in the global attention case with the main difference that we have the attention only on $h^T_{t_{s-1} + 1}$. The whole segmental model is depicted in Figure 2.

### 3.1. Label model variations

The label model is depicted in Figure 3 and defined as

$$p(a_s \mid \ldots) = (\text{softmax} \circ \text{Linear} \circ \text{maxout} \circ \text{Linear}) \left( \text{LSTM}(c^s_{t_s-1}, a^s_{t_s-1}, c_s) \right)$$

$$c_s = \sum_{t = t_{s-1} + 1}^{t_s} \alpha_{s,t} \cdot h_t$$

$$\alpha_{s,t} = \frac{\exp(e_{s,t})}{\sum_{\tau=t_{s-1} + 1}^{t_s} \exp(e_{s,\tau})}$$

$$e_{s,t} = (\text{Linear} \circ \text{tanh} \circ \text{Linear}) \left( \text{LSTM}(c^s_{t_s-1}, a^s_{t_s-1}, h_t) \right).$$

The attention weights here are only calculated on the interval $[t_{s-1} + 1, t_s]$. Further, we do not have attention weight feedback as there is no overlap between the segments. Otherwise the model is exactly the same as the global attention decoder, to allow for direct comparisons, and also to import model parameters.

Another variation is on the dependencies. In any case, we depend on the full label history $a^S_{t_{s-1}}$. When we have the dependency on $c^s_{t_{s-1}}$ as it is standard for LSTM-based attention models, this implies an implicit dependency on the whole past segment boundaries $t^S_{s-1}$, which removes the option of an exact first-order dynamic programming implementation for forced alignment or exact sequence-likelihood training. So, we also test the variant where we remove the $c^s_{t_{s-1}}$ dependency in the equations above.

### 3.2. Silence modeling

The output label vocabulary of the global attention model usually does not include silence as this is not necessarily needed. Our segments completely cover the input sequence, and thus the question arises whether the silence parts should be separate segments or not, i.e. whether we should add silence to the vocabulary. Additionally, as silence segments tend to be longer, we optionally split them up. We will perform experiments for all three variants, also for the global attention model.

Further, when we include silence, the question is whether this should be treated just as another output label, or treated in a special way, e.g. separating it from the softmax. We test...
some preliminary variations on this.

3.3. Length model variations and length normalization

No length model. The simplest variant is without any length model (α = 0). We might argue that this is closest to the global attention model. In this case, we use length normalization instead during recognition as explained below, just like we do for the global attention model.

Static length model. Here we estimate a model

\[
p(t_s = t_{s-1} + \Delta t \mid a_s, t_{s-1}) = \begin{cases} 
0, & 1 \leq \Delta t \leq \delta_{\text{max}} \\
\exp(-|\mu_{a_s} - \Delta t|)/Z, & \text{otherwise} 
\end{cases}
\]

with mean segment length \( \mu \), maximum segment length \( \delta_{\text{max}} \) and normalization \( Z \), where \( \mu \) is estimated based on some given alignments. Note that this invalidates a proper probability normalization when combined with the label model \( p(a_s|...) \). However, we can interpret this as a log-linear combination of the joint model \( p(t_s, a_s|...) \), although we do not use a proper renormalization here.

Neural length model. This model is defined as

\[
p(t_s = t \mid a_{s-1}^s, t_{s-1}^s, h_T^s) = \begin{cases} 
0, & t < t_{s-1} + 1 \\
\left[ \prod_{t'=t_{s-1}+1}^{t} 1 - q_{t'} \right] \cdot q_t, & t \geq t_{s-1} + 1 
\end{cases}
\]

\[q_{t'} = (\text{sigmoid} \circ \text{Linear} \circ \text{tanh} \circ \text{LSTM})(\omega_{t'}^{-1}, h_{t'}^s)\]

The \( q_{t'} \) model here is defined in a framewise manner and predicts whether the new segment ends in the frame \( t' \), similar as in \[10,37,38,40,47,49\]. This \( q_{t'} \) model gets in the framewise alignment \( \omega_{t'}^{-1} \) so far, where we encode the labels \( a_{s-1}^s \) at the segment boundaries \( t_{s-1}^s \) and encode a special blank symbol \( b \) otherwise, specifically

\[\omega_{t'} = \begin{cases} a_{s'}, & t_{s'} = t'' \\
(b), & \text{otherwise} \end{cases}\]

3.4. Training

For simplicity, we use framewise cross entropy (CE) as training criterion using a given framewise alignment. Other from-scratch training schemes are possible without the requirement of a given alignment but we do not investigate this further here. We use alignments from a HMM for the variants with silence and alignments from a RNA model \[51\] otherwise. This defines the segment boundaries \( t_s \). We minimize the loss

\[L(\theta) = \sum_{s=1}^{S} -\log p_\theta(t_s | ... ) - \log p_\theta(a_s | ... )\]

w.r.t. the model parameters \( \theta \) over the training data.

For the segmental-attention model, we can also directly use the model parameters of a global-attention model, as it is just the same model except that the attention is limited to the segment, and expect the length model. We also perform such experiments.

3.5. Recognition: Time-sync. decoding & pruning

For the segmental model, our decision rule is

\[x_1^{T*} \mapsto \hat{S}, \hat{a}_1^S, \hat{t}_1^S := \arg\max_{S,a_1^S,t_1^S} \left( \sum_{s=1}^{S} \alpha \cdot \log p_{\theta}(t_s | ... ) + \log p_{\theta}(a_s | ... ) \right) \cdot S^{-\gamma}.\]

We test different length model scales \( \alpha \) and we either enable or disable length normalization (\( \gamma = 0, 1 \)) \[52\]. We don’t use an external language model in this work.

We use time synchronous decoding, meaning that the state extension and pruning is synchronized in a frame-by-frame manner, i.e., all active hypotheses are in the same time frame. We have two separate decoder implementations for our segmental model, both of them being time-synchronous but differing in the pruning variant. They are both based on our time-synchronous transducer decoding and use a reformulation of the segmental model as a transducer model similar to \[47\]. The label model contribution is added at the end of a segment. In case of the neural length model, we can use \( q \) for a framewise contribution, or otherwise add the length model contribution at the end of a segment.

Segmental search. At each time frame, we only prune those hypotheses reaching segment end against each other, while others are kept for further extension unless they reach a maximum segment length \( \delta_{\text{max}} \). We use Viterbi recombination of hypotheses with the same label history. This implementation is more complex and runs on CPU.

Simple search. At each time frame, we prune all hypotheses together regardless of reaching segment end or not. This is done only with the neural length model, where ongoing segments still have an intermediate length probability (framewise \( q_t \)) for pruning. This is exactly the transducer search without any recombination. The algorithm is simple and we have an efficient GPU-based implementation.

This is in contrast to label-synchronous search as it is done usually for global attention models or also other segmental models \[46\], where all current active hypotheses have the same number of output labels.

4. EXPERIMENTS & ANALYSIS

We perform all our experiments on the Switchboard 300h English telephone speech corpus \[53\]. We use a vocabulary of 1030 byte pair encoding (BPE) \[54\] tokens with optional silence. The encoder is always a 6-layer bidirectional LSTM encoder of 1024 dimensions in each direction with downsampling factor 6 via max-pooling and has 137M parameters. The decoder uses a single LSTM of 1024 dimensions and has 24M parameters without the length model and is mostly identical for global-attention and segmental-attention. The length model has 1.3M parameters. We implement the neural models and the simple search for recognition in RETURNN \[55\]
Table 1: Global vs. segmental attention without length model but length normalization. Word-error-rate (WER) on Hub5’00 with substitutions (S), deletions (D) and insertions (I), and search errors (SE). All models with silence; segmental search for segmental models.

<table>
<thead>
<tr>
<th>Model</th>
<th>Parameters</th>
<th>WER (%)</th>
<th>SE (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>S</td>
<td>D</td>
</tr>
<tr>
<td>Global</td>
<td>Trained</td>
<td>9.3</td>
<td>4.6</td>
</tr>
<tr>
<td>Segmental</td>
<td>From global</td>
<td>6.7</td>
<td>35.8</td>
</tr>
<tr>
<td></td>
<td>Trained</td>
<td>6.4</td>
<td>17.2</td>
</tr>
</tbody>
</table>

Table 2: The model scores $\log p(a_s|\ldots)$ (higher is better) for the global and segmental attention models over the ground truth and recognized sequence per label and in total for an example sequence.

<table>
<thead>
<tr>
<th>Labels</th>
<th>Ground-truth (= recognized)</th>
<th>Segmental attention</th>
</tr>
</thead>
<tbody>
<tr>
<td>[Silence]</td>
<td>−0.11</td>
<td>−0.11</td>
</tr>
<tr>
<td>[Silence]</td>
<td>−0.10</td>
<td>−0.10</td>
</tr>
<tr>
<td>[Silence]</td>
<td>−0.11</td>
<td>−0.11</td>
</tr>
<tr>
<td>you</td>
<td>−0.09</td>
<td>−0.06</td>
</tr>
<tr>
<td>know</td>
<td>−0.25</td>
<td>−0.10</td>
</tr>
<tr>
<td>gra@@</td>
<td>−0.06</td>
<td>−0.31</td>
</tr>
<tr>
<td>de</td>
<td>−0.13</td>
<td>−0.02</td>
</tr>
<tr>
<td>school</td>
<td>−0.02</td>
<td>−0.06</td>
</tr>
<tr>
<td>or</td>
<td>−0.23</td>
<td>−0.01</td>
</tr>
<tr>
<td>so</td>
<td>−0.10</td>
<td>−0.12</td>
</tr>
</tbody>
</table>

$\Sigma/S$: −1.71, −1.44, −0.54

$\Sigma/S$: −0.11, −0.10, −0.06

Fig. 4: Top: Attention weights of the global attention model. Bottom: Segment boundaries and attention weights of the segmental model. We also show the reference HMM alignment.

Table 3: Comparing different length models, with optional length normalization. Segmental search; models with silence.

<table>
<thead>
<tr>
<th>Length model</th>
<th>Length norm.</th>
<th>WER (%)</th>
<th>SE (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td>No</td>
<td>5.3</td>
<td>29.7</td>
</tr>
<tr>
<td>Static</td>
<td>Yes</td>
<td>6.4</td>
<td>17.2</td>
</tr>
<tr>
<td>Neural</td>
<td>No</td>
<td>11.9</td>
<td>18.9</td>
</tr>
<tr>
<td></td>
<td>Yes</td>
<td>10.2</td>
<td>16.3</td>
</tr>
<tr>
<td></td>
<td>Yes</td>
<td>9.0</td>
<td>6.2</td>
</tr>
<tr>
<td></td>
<td>Yes</td>
<td>9.1</td>
<td>5.9</td>
</tr>
</tbody>
</table>

We see that restricting the attention to the segment actually improves the recognition score slightly ($-0.10 > -0.11$). However, now the segmental models recognize another bad sequence with even better scores ($-0.06$). So the score of the correct sequence improved but the score of incorrect sequences improved even more.

We also looked at the segment boundaries of the recognized sequence. The segment boundaries together with the attention weights can be seen in Figure 4. We notice that the segment boundaries are very off and they often cover multiple labels which explains the high deletion errors from Table 1. This observation is consistent over many other examples as well. We also make the same observations when we train the segmental model in contrast to importing the global attention model parameters.

To summarize: the label model yields good scores even for bad segment boundaries. This is probably due to the attention mechanism. This demonstrates that we need some sort of length model to get better segment boundaries.

4.2. Length model comparison

In all following experiments, we always train the segmental model and do not import weights.

We compare our different length models and length normalization in Table 3. As expected, without any length model, the length normalization improves the performance. For the other length models, we use scale $\alpha = 1$. We notice that the

and the segmental search in RASR [56]. All the code and recipes can be found online.

We also provide the number of search errors by counting the number of sequences where the ground truth sequence has a higher score than the recognized sequence, as a sanity check to indicate how much errors we make due to the approximations in the search.

4.1. Failures when no length model is used

Some initial results comparing global and segmental attention are shown in Table 1. This is without a length model but with length normalization. For better comparison, all models include silence labels. The segmental models perform badly. In the following, we analyze why that is.

As a first step, we want to understand the behavior when we take over the parameters of a global attention model and just restrict the attention on given segment boundaries. We now look at individual scores per label. We chose a sequence where the global attention model recognizes exactly the ground truth label sequences but the segmental model recognizes some different sequence. The scores are in Table 2.

simple static length model yields even worse results than no length model. We analyze that in the following subsection and a tuned \( \alpha \) will improve that. Overall, the neural length model yields the best results, where the length normalization has a negligible effect. In better segmental models with neural length model as presented later (e.g. Table 10), length normalization was hurtful and we did not use it. To conclude, length normalization is only needed without or with a weak length model.

### 4.3. Failures with static length model

We want to better understand why the static length model fails and whether it fails for similar reasons as without any length model. We again look into the model scores for an example sequence in Table 4. Now we notice a new problem: The length model scores dominate much over the label scores. The label model scores of the ground-truth sequence is actually better than for the recognized sequence. However, the length model dominates and prefers some incorrect segment boundaries.

To overcome this, we test different length model scales \( \alpha \) in Table 5, to transition from no length model to the static length model. We see that we can still improve over the cases \( \alpha = 0 \) or \( \alpha = 1 \) but are still behind the global attention model. To summarize: A weak length model also leads to suboptimal results. Thus we will focus now on the segmental model with neural length model.

### 4.4. Silence variants

We compare the standard global-attention case of not having silence vs. having silence, either being split up into multiple segments in case they are too long, or not split up. The information on silence is extracted from our HMM alignment.

| Table 4: Segmental model scores using static length model: Label scores \( \log p(\mathbf{x}_s|\ldots) \) and static length model \( \log p(t_s|\ldots) \).
<table>
<thead>
<tr>
<th>Labels</th>
<th>Label model scores</th>
<th>Length model scores</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Ground-truth</td>
<td>Recognized</td>
</tr>
<tr>
<td>[Silence]</td>
<td>(-0.11)</td>
<td>(-0.11)</td>
</tr>
<tr>
<td>[Silence]</td>
<td>(-0.10)</td>
<td>(-0.10)</td>
</tr>
<tr>
<td>parti@@</td>
<td>(-0.10)</td>
<td>(-0.08)</td>
</tr>
<tr>
<td>cu@@</td>
<td>(-0.05)</td>
<td>(-0.04)</td>
</tr>
<tr>
<td>lar@@</td>
<td>(-0.06)</td>
<td>(-1.18)</td>
</tr>
<tr>
<td>ly</td>
<td>(-0.05)</td>
<td>(-0.14)</td>
</tr>
<tr>
<td>if</td>
<td>(-0.02)</td>
<td>(Deletion)</td>
</tr>
<tr>
<td>you’re</td>
<td>(-0.61)</td>
<td>(-1.60)</td>
</tr>
<tr>
<td>h@</td>
<td>(-0.27)</td>
<td>(-0.54)</td>
</tr>
<tr>
<td>un@</td>
<td>(-0.06)</td>
<td>(-0.31)</td>
</tr>
<tr>
<td>g@</td>
<td>(-0.01)</td>
<td>(-0.20)</td>
</tr>
<tr>
<td>ry</td>
<td>(-0.03)</td>
<td>(-0.05)</td>
</tr>
<tr>
<td>(\Sigma)</td>
<td>(-1.75)</td>
<td>(-2.90)</td>
</tr>
</tbody>
</table>

| Table 5: Static length model scales \( \alpha \)
<table>
<thead>
<tr>
<th>Length model scale</th>
<th>0.0</th>
<th>0.01</th>
<th>0.1</th>
<th>0.3</th>
<th>1.0</th>
</tr>
</thead>
<tbody>
<tr>
<td>WER (%)</td>
<td>25.3</td>
<td>23.9</td>
<td>21.8</td>
<td>24.9</td>
<td>31.7</td>
</tr>
</tbody>
</table>

The results are in Table 6. The variant without silence is better in all cases. Without silence, now the segmental attention model performs even better than the global attention model. We think that in principle it should be possible to improve the variant with silence and we have some ongoing work on this.

### 4.5. Search comparison

Results of our two search implementations as described in Section 3.5 are collected in Table 7, comparing the same segmental model. Without silence, we get better performance with the segmental search. With silence, the result is unexpected. We can see that segmental search even has less search errors, so it looks like the model is worse but this needs further investigation, as also already mentioned in the last subsection. Segmental search is slower than simple search, although the numbers are not directly comparable due to CPU vs. GPU. Compared to the global attention model, for the segmental search, the computational complexity increases by a constant factor by the maximum segment length \( \delta_{\text{max}} \), but the attention computation complexity decreases from \( T \cdot S \) to \( \delta_{\text{max}} \cdot S \), which dominates once \( T \) becomes big.

### 4.6. State vector type

As described in Section 3.1, we have two variants of the state vector, either with dependency on \( c_{t-1} \) or without. Results are in Table 8. As expected, the global attention model benefits a lot by having this dependency, as it can use it to better

| Table 6: Comparing silence variants.
<table>
<thead>
<tr>
<th>Model</th>
<th>Silence variant</th>
<th>WER (%)</th>
<th>SE (%)</th>
<th>Time (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Global</td>
<td>None</td>
<td>9.2</td>
<td>3.3</td>
<td>2.2</td>
</tr>
<tr>
<td></td>
<td>No split</td>
<td>9.5</td>
<td>3.4</td>
<td>2.1</td>
</tr>
<tr>
<td></td>
<td>Split</td>
<td>9.3</td>
<td>4.0</td>
<td>2.0</td>
</tr>
<tr>
<td>Segmental</td>
<td>None</td>
<td>8.7</td>
<td>3.5</td>
<td>1.9</td>
</tr>
<tr>
<td></td>
<td>Split</td>
<td>9.0</td>
<td>6.2</td>
<td>1.9</td>
</tr>
</tbody>
</table>

| Table 7: Search comparison of simple search vs. segmental search, using segmental models with silence or without. *Not directly comparable because of CPU vs. GPU.
<table>
<thead>
<tr>
<th>Variant</th>
<th>Simple search</th>
<th>Segmental search</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>WER (%)</td>
<td>SE (%)</td>
</tr>
<tr>
<td>W/o sil.</td>
<td>15.0</td>
<td>0.8</td>
</tr>
<tr>
<td>W/ sil.</td>
<td>16.4</td>
<td>2.4</td>
</tr>
</tbody>
</table>

| Table 8: Testing state vector dependency on \( c_{t-1} \). Segmental model uses segmental search, neural length model.
<table>
<thead>
<tr>
<th>Model</th>
<th>( c_{t-1} ) dependency</th>
<th>WER (%)</th>
<th>SE (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Global</td>
<td>Yes</td>
<td>15.3</td>
<td>0.5</td>
</tr>
<tr>
<td></td>
<td>No</td>
<td>16.8</td>
<td>0.5</td>
</tr>
<tr>
<td>Segmental</td>
<td>Yes</td>
<td>14.5</td>
<td>0.0</td>
</tr>
<tr>
<td></td>
<td>No</td>
<td>14.9</td>
<td>0.0</td>
</tr>
</tbody>
</table>
Table 9: Test generalization on longer sequences, concatenating $C$ consecutive sequences in the Hub5’00 corpora.

<table>
<thead>
<tr>
<th>$C$</th>
<th>Seq. length (secs)</th>
<th>WER (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>mean ± std</td>
<td>min-max</td>
</tr>
<tr>
<td>1</td>
<td>2.91 ± 2.57</td>
<td>0.08 – 15.54</td>
</tr>
<tr>
<td>2</td>
<td>8.10 ± 5.19</td>
<td>0.27 – 63.97</td>
</tr>
<tr>
<td>4</td>
<td>18.28 ± 9.64</td>
<td>0.27 – 100.46</td>
</tr>
<tr>
<td>10</td>
<td>46.65 ± 22.52</td>
<td>0.33 – 198.10</td>
</tr>
<tr>
<td>20</td>
<td>87.41 ± 44.83</td>
<td>0.89 – 282.78</td>
</tr>
</tbody>
</table>

Table 10: Overall comparison, comparing global attention models, transducers and segmental models from the literature and our final best models, different number of epochs (#Ep), seg. model uses length model scales 1.0 or 0.7.

<table>
<thead>
<tr>
<th>Work</th>
<th>Model</th>
<th>Type</th>
<th>#Ep</th>
<th>L.M.</th>
<th>Hub5’00</th>
<th>WER (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Gl.Att.</td>
<td>BPE</td>
<td>33</td>
<td>30</td>
<td>600</td>
<td>44</td>
</tr>
<tr>
<td></td>
<td>Transd.</td>
<td>BPE</td>
<td>1k</td>
<td>70</td>
<td>100</td>
<td>56</td>
</tr>
<tr>
<td>Ours</td>
<td>Seg/7/0</td>
<td>BPE</td>
<td>25</td>
<td>50</td>
<td>800</td>
<td>282</td>
</tr>
</tbody>
</table>

4.7. Generalization on longer sequences

We investigate performance on longer sequences than seen during training by simply concatenating $C$ consecutive sequences during recognition. We compare the global attention model to the segmental attention model and collect results in Table 9. The segmental model generalizes much better than global attention model. We also observe that the segmental model with explicit silence generalizes slightly better than without explicit silence. We believe that segmental models would generalize even better to very long sequences if they would have seen some sentence concatenation in training. Then, they should even improve the WER the longer the sequence becomes due to more context, as has been seen for language modeling [6, 57].

4.8. Overall comparison

We collect our final results comparing our best global-attention model to our best segmental-attention model and other results from the literature in Table 10. Our segmental-attention model still uses an LSTM acoustic model, no external language model yet, and is not trained as long as other results which explains the gap to some other work from the literature. We see that segmental-attention overall performs slightly better than global-attention.

5. CONCLUSIONS

We introduced a novel type of segmental-attention models derived from the global-attention models. Our investigation of the modeling of the segment boundaries demonstrates and explains why a good length model is important for such models: the label model on its own still yields good scores even for bad segment boundaries. This can be explained due to the flexibility of attention but similar observations has also been made for other segmental models [40, 41]. This is in contrast to the experience of HMMs were the time-distortion penalties are less important.

In the end, our final segmental-attention model improves over the global-attention model, while at the same time satisfies all our motivations, namely it is monotonic, allows for online-streaming, potentially more efficient (at least the neural model, the search is currently slower but this is just for technical implementation reasons and can be improved), and it generalizes much better to long sequences. We think the segmental model can still be improved by not using $[t_{s}−1 + t_{s}]$ as the attention window but instead some fixed-size window where $t_{s}$ is the center position.

When looking at the equivalence of transducer and segmental models [47], we note that the main difference of our segmental model to common transducer models is the use of attention on the segment, while otherwise both models are conceptually similar.

6. ACKNOWLEDGEMENTS

This work was partially supported by NeuroSys which, as part of the initiative “Clusters4Future”, is funded by the Federal Ministry of Education and Research BMBF (03ZU1106DA); a Google Focused Award. The work reflects only the authors’ views and none of the funding parties is responsible for any use that may be made of the information it contains.

7. REFERENCES


[45] Tamer Alkhouli, Alignment-Based Neural Networks for Machine Translation, Ph.D. thesis, RWTH Aachen University, Computer Science Department, RWTH Aachen University, Aachen, Germany, July 2020.


