Do end-to-end speech recognition models care about context?

Borgholt, Lasse; Havtorn, Jakob D.; Agic, Željko; Søgaard, Anders; Maaløe, Lars; Igel, Christian

Published in:
Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH

DOI:
10.21437/Interspeech.2020-1750

Publication date:
2020

Document version
Publisher's PDF, also known as Version of record

Document license:
Other

Citation for published version (APA):
Do End-to-End Speech Recognition Models Care About Context?

Lasse Borgholt\textsuperscript{1,2}, Jakob D. Havtorn\textsuperscript{2}, Željko Agić\textsuperscript{2}, Anders Søgaard\textsuperscript{1}, Lars Maaløe\textsuperscript{2}, Christian Igel\textsuperscript{1}

\textsuperscript{1}Department of Computer Science, University of Copenhagen, Denmark
\textsuperscript{2}Corti, Copenhagen, Denmark

\{borgholt, sogaard, igel\}@di.ku.dk, \{lb, jdh, za, lm\}@corti.ai

Abstract

The two most common paradigms for end-to-end speech recognition are connectionist temporal classification (CTC) and attention-based encoder-decoder (AED) models. It has been argued that the latter is better suited for learning an implicit language model. We test this hypothesis by measuring temporal context sensitivity and evaluate how the models perform when we constrain the amount of contextual information in the audio input. We find that the AED model is indeed more context sensitive, but that the gap can be closed by adding self-attention to the CTC model. Furthermore, the two models perform similarly when contextual information is constrained. Finally, in contrast to previous research, our results show that the CTC model is highly competitive on WSJ and LibriSpeech without the help of an external language model.

Index Terms: automatic speech recognition, end-to-end speech recognition, connectionist temporal classification, attention-based encoder-decoder

1. Introduction

Connectionist temporal classification (CTC) \cite{1} and attention-based encoder-decoder (AED) models \cite{2, 3} are arguably the most popular choices for end-to-end automatic speech recognition (E2E ASR). However, it has been unclear if CTC and AED models process speech in qualitatively different ways. The use of sentence-level context is important for human speech perception \cite{4}, but has not been studied for ASR. Previous research has claimed that AED models learn a better implicit language model given enough training data \cite{5}. Furthermore, comparisons of the two models have suggested that CTC models are inferior without the help of an external language model \cite{5, 6}, which leads to the hypothesis that CTC models are incapable of exploiting long temporal dependencies.

We study how the two E2E ASR models utilize temporal context. For this purpose, we consider first-order derivatives \cite{7, 8} and the occlusion of input features \cite{9}. While these methods have been frequently used to analyze natural language processing models \cite{10, 11, 12, 13} their application to speech recognition has been limited \cite{14, 15}.

We first highlight three general architectural differences between the two approaches and argue that these may enable AED models to utilize more temporal context than CTC models (section 2). Further, we define an intrinsic measure of context sensitivity based on the partial derivative of individual character predictions with respect to the input audio (section 3.1 and figure 1). This allows us to analyze the sensitivity across the temporal dimension of the input for any E2E ASR model. Finally, we devise an experiment to directly compare model performance when context is constrained (section 3.2). For this, we use hand-annotated word-alignments to accurately occlude temporal context. Our contributions are as follows:

1. Through a derivative-based sensitivity analysis we show that the AED model is more context sensitive than the CTC model. Our ablation study attributes this difference to the attention-mechanism which closes the gap when applied to the CTC model.

2. Although the AED model is more context sensitive than the CTC model without an attention-mechanism, we find that the two models perform similarly when contextual information is constrained by occluding surrounding words in the input audio.

3. In contrast to previous comparisons, we show that the CTC model is highly competitive with the AED model without the help of an external language model. Using a deep and densely connected architecture, both models reach a new E2E state-of-the-art on the WSJ task.

![Figure 1: Sensitivity scores for the character “p” in the correctly predicted sentence “who is going to stop me” by the CTC model trained on LibriSpeech. Hand-annotated word and phone alignments from the TIMIT dataset are shown in the bottom. The temporal spans corresponding to different levels of accumulated sensitivity are shown in the top. By averaging these across all non-blank character predictions in a test set, we obtain a measurement of the model’s context sensitivity.](http://dx.doi.org/10.21437/Interspeech.2020-1750)
2. End-to-End Speech recognition

2.1. Connectionist Temporal Classification Models

Given a sequence of real valued input vectors \( x = (x_1, \ldots, x_L) \), CTC models compute an output sequence \( \hat{y} = (\hat{y}_1, \ldots, \hat{y}_T) \), where each \( \hat{y}_k \) is a categorical probability distribution over the target character set. Apart from the letters a-z, white-space and apostrophe (‘), the character set also includes the special blank token (\( \_ \)). The input and output lengths, \( T \) and \( U \), are related by \( U = \lceil \frac{T}{R} \rceil \) where \( R \) is a constant reduction factor achieved by striding or stacking adjacent temporal representations. In this study, we never use an external language model. Instead, we rely on a simple greedy decoder \( \beta(\cdot) \) that collapses repeated characters and removes blank tokens (e.g., ’cat’ \( \rightarrow \) cat). The \( \beta(\cdot) \) function operates on the predicted alignment path \( \hat{q} = (\hat{q}_1, \ldots, \hat{q}_U) \) obtained by letting \( \hat{q}_u = \arg \max_{y_u} \hat{y}_u \cdot y_u \).

This decoding mechanism results from the CTC loss function. The loss is computed by summing the probability of all alignment paths \( q = (q_1, \ldots, q_U) \) that translate to the target sequence \( y \). The probability of a single path is given by:

\[
P(q|x) = \prod_{u=1}^{U} \hat{y}_{u \cdot q_{u}}
\]

Given the set of paths \( \{ q | \beta(q) = y \} = \beta^{-1}(y) \) that translate to a given target transcript, the total probability is:

\[
P(y|x) = \sum_{q \in \beta^{-1}(y)} P(q|x)
\]

The loss is simply \( L(y, \hat{y}) = \ln(P(y|x)) \) which can be computed efficiently with dynamic programming [1].

2.2. Attention-based Encoder-Decoder Models

AED models first encode the input \( x \) to a sequence of vectors \( h = (h_1, \ldots, h_L) = \text{ENCODER}(x) \) which is passed to an autoregressive decoder function \( \text{DECODER}(\cdot) \). We reuse \( U \) to denote the length of \( h \) to emphasize that, as with CTC models, it is defined by a constant reduction factor \( R \). However, AED models are typically robust to a high reduction factor \( R \leq 2^4 \) compared to CTC models \( R \leq 2^2 \) [5]. Operating at a lower temporal resolution should make it easier for recurrent encoder layers (section 2.3) to pass information across longer time spans.

![Diagram of encoder architecture used for both CTC and AED models. * Only applied in the bottleneck layer of the first dense LSTM block for the AED model to achieve \( R = 4 \).]

Roughly speaking, we could write the decoder as \( \hat{y}_k = \text{DECODE}(h, \hat{y}_{k-1}, s_{k-1}, a_{k-1}) \) where \( \hat{y}_k \) is a probability distribution over characters, \( s_k \) is the decoder state and \( a_k \) is the attention vector. Unlike CTC models, there are no repeated characters or blank tokens to interleave the final predictions. Thus, given the same output sequence, we have \( K \leq U \). As with the encoder, lower temporal resolution between decoder steps could make it easier to pass information between predictions.

Emphasizing more detail, we split the \( \text{DECODER}(\cdot) \) function into the following sequence of computations:

\[
s_k = \text{RECURRENT}(s_{k-1}, [\Phi(\hat{y}_{k-1}); a_{k-1}]) \tag{3}
\]

\[
a_k = \text{ATTEND}(s_k, h) \tag{4}
\]

\[
\hat{y}_k = \text{PREDICT}(a_k) \tag{5}
\]

Here \( [\cdot; \cdot] \) denotes the concatenation of two vectors and \( \Phi(\cdot) \) is a non-differentiable embedding lookup.\(^1\) The \( \text{RECURRENT}(\cdot) \) function can take the form of any recurrent neural network architecture. We use a single LSTM [16] cell for all our experiments. The \( \text{PREDICT}(\cdot) \) function is a single fully-connected layer followed by the softmax function. The following steps define the \( \text{ATTEND}(\cdot) \) function:

\[
e_{k,u} = v^\top \tanh(W_s s_k + W_h h_u) \tag{6}
\]

\[
\alpha_{k,u} = \exp(e_{k,u}) \tag{7}
\]

\[
c_k = \sum_{u=1}^{U} \alpha_{k,u} h_u \tag{8}
\]

\[
a_k = \tanh(W_v [c_k; s_k]) \tag{9}
\]

Where \( v \), \( W_s \), \( W_h \) and \( W_v \) are trainable parameters. The computation of the energy coefficient \( e_{k,u} \) is taken from [17]. Note that each energy coefficient, and thus each attention weight \( \alpha_{k,u} \), is computed identically for all encoder representations \( h_u \). Unlike recurrent network connections, combining information from time steps far apart does not require propagating the information through a number of computations proportional to the distance between the time steps.

Thus, we have highlighted three components that could make AED models more context sensitive: (I) Encoder resolution, (II) decoder resolution and (III) the attention-mechanism.

2.3. Encoder Architecture

Whereas the main contribution of the CTC framework is the loss function, the AED model relies on a more complex architecture that allows it to be trained with a simple cross-entropy loss. To see this, note that the CTC forward pass can be stated as a subset of the functions introduced in section 2.2:

\[
h = \text{ENCODER}(x) \tag{10}
\]

\[
\hat{y}_u = \text{PREDICT}(h_u) \tag{11}
\]

As in previous work, we use convolutions followed by a sequence of bidirectional recurrent neural networks [18, 19, 20]. Our final encoder has 10 bidirectional LSTM layers with skip-connections inspired by [21]. The outputs of the forward and backward cells are summed after each LSTM layer. Default is \( R = 2 \) for CTC and \( R = 4 \) for AED. See figure 2.

\(^1\)The lookup is not captured by the gradient-based sensitivity analysis presented in 3.1.
We used two different approaches for analyzing temporal context utilization of the two E2E ASR models. The derivative-based sensitivity analysis (3.1) can be used to compare a set of models on any dataset. However, as we will see, there is no guarantee that the differences found with this approach translate to better performance. The occlusion-based analysis (3.2) allows us to evaluate how the models respond when we remove temporal context. This measure is easy to interpret and can be used to directly assess the importance of temporal context, but requires hand-annotated word-alignments which are rarely available in publicly available datasets.

3.1. Derivative-based Sensitivity Analysis

We define a sequence of sensitivity scores $r_k$ ($r_k$ for CTC models) across the temporal dimension of the input space for each predicted character. Let $F$ be the number of spectral input features and $Q$ the size of the output character set:

$$r_{k,t} = \sum_{q=1}^{Q} \sum_{f=1}^{F} \frac{\partial \hat{y}_{k,q}}{\partial x_{f,t}}$$

An example is shown in figure 1. Our goal is to measure the dispersion of these scores across the input time steps. We do so by summing the scores from largest to smallest and measuring the temporal span of the scores accumulated for a certain percentage of the total sensitivity. For example, the set of scores needed to account for 10% of the total sensitivity may be $\{r_{k,3}, r_{k,7}, r_{k,8}, r_{k,10}\}$.

3.2. Occlusion-based Analysis

To directly test the dependence on contextual information, we use hand-annotated word-alignments to systematically occlude context. Given a word $w_t$, we test how well a model recognizes the word given different levels of context. That is, we crop out the audio segment corresponding to $w_{t-C}, ..., w_{t+C}$ where $C$ is the maximum number of context words visible on each side. If the target word $w_t$ is in the predicted sequence, we accept the hypothesis. To avoid ambiguous situations where the target word is identical to one of the $2C$ context words, we only make use of sentences that consist of a sequence of unique words.

### Table 1: Word error rates on the clean and other test sets of LibriSpeech. None of the above use an external language model.

<table>
<thead>
<tr>
<th>Model</th>
<th>clean</th>
<th>other</th>
<th>type</th>
<th>params</th>
</tr>
</thead>
<tbody>
<tr>
<td>Li et al., 2019 [22]</td>
<td>3.86</td>
<td>11.95</td>
<td>CTC</td>
<td>333 M</td>
</tr>
<tr>
<td>Kim et al., 2019 [23]</td>
<td>3.66</td>
<td>12.39</td>
<td>AED</td>
<td>~320 M</td>
</tr>
<tr>
<td>Park et al., (2019) [18]</td>
<td>2.80</td>
<td>6.80</td>
<td>AED</td>
<td>~280 M</td>
</tr>
<tr>
<td>Irie et al., (2019) [24]</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Small - Grapheme</td>
<td>7.9</td>
<td>21.3</td>
<td>AED</td>
<td>7 M</td>
</tr>
<tr>
<td>Small - Word-piece</td>
<td>6.1</td>
<td>16.4</td>
<td>AED</td>
<td>20 M</td>
</tr>
<tr>
<td>Medium - Grapheme</td>
<td>5.6</td>
<td>15.8</td>
<td>AED</td>
<td>35 M</td>
</tr>
<tr>
<td>Medium - Word-piece</td>
<td>5.0</td>
<td>14.1</td>
<td>AED</td>
<td>60 M</td>
</tr>
<tr>
<td><strong>Our work:</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Deep LSTM</td>
<td>5.13</td>
<td>16.03</td>
<td>CTC</td>
<td>17.7 M</td>
</tr>
<tr>
<td>Deep LSTM</td>
<td>5.45</td>
<td>17.05</td>
<td>AED</td>
<td>19.8 M</td>
</tr>
</tbody>
</table>

### Table 2: Word error rates on the eval92 test set of WSJ. None of the above use an external language model.

<table>
<thead>
<tr>
<th>Model</th>
<th>eval92</th>
<th>type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chan et al., (2016) [26]</td>
<td>9.6</td>
<td>AED</td>
</tr>
<tr>
<td>Labour et al., (2018) [27]</td>
<td>9.3</td>
<td>AED</td>
</tr>
<tr>
<td><strong>Our work:</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Deep LSTM</td>
<td>9.25</td>
<td>CTC</td>
</tr>
<tr>
<td>Deep LSTM</td>
<td>9.25</td>
<td>AED</td>
</tr>
</tbody>
</table>
4.3. CTC vs. AED

As hypothesized, figure 3.1 and 3.2 reveal that our AED models utilized a larger temporal context than the CTC models based on the sensitivity scores. The trend was consistent across all levels of accumulated sensitivity scores. In figure 3.3, we see the same pattern when evaluated on the TIMIT dataset which will be used for the occlusion-based analysis.

4.4. Temporal resolution

We trained the AED model with three different temporal encoder resolutions \( R = 2, 4, 8 \) on the WSJ dataset. \( R \) was configured by increasing stride in each of the three convolutional layers. As seen in figure 3.4, encoder resolution had no impact on context sensitivity.

To test decoder resolution, we interleaved the target transcript with one or two redundant blank tokens to effectively increase the target length to \( K \cdot 2 \) or \( K \cdot 3 \). Figure 3.4 shows that decoder resolution had no impact on context sensitivity.

4.5. Attention-mechanism

To test how the attention-mechanism affects context sensitivity, we incorporated the \( \text{ATTEND}(\cdot) \) function in the CTC architecture. Instead of passing \( h_u \) directly to \( \text{PREDICT}(\cdot) \), we first applied self-attention:

\[
\hat{y}_u = \text{PREDICT}(\text{ATTEND}(h_u, h)) \quad (13)
\]

We trained this model on the WSJ dataset and compared it to the AED model and the CTC model without attention in figure 3.5. The attention-mechanism closed the gap in context sensitivity between the two models. Thus, the difference found in sections 4.3 is likely a result of this architectural component that can be easily incorporated in a CTC model. However, a large \( U \) results in high memory consumption. Therefore, we used a smaller model where the two dense LSTM blocks are replaced by three LSTM layers with 128 units for the experiments shown in figures 3.4 and 3.5.

4.6. Occlusion performance

Figure 3.6 shows how model performance is affected under different context constraints. We see that both the CTC and AED model suffered severely when contextual information was completely removed. The models came close to optimal performance when approximately three words were allowed on each side of the target word. Thus, temporal context is an important factor for both models. This result aligns well with the common n-gram size (3–4) when decoding with the help of a statistical language model [25, 35, 36].

Based on the results in section 4.3, we would expect that the AED models rely more on the temporal context than the CTC model. However, we do not see such a trend in figure 3.6. Indeed, there was no pronounced or consistent difference between the two models regardless of training source. This result implies that the architectural differences between the AED and CTC models do not necessarily translate to a performance difference. It may be that the AED model included more evidence from context than the CTC model, but the results in figure 3.6 indicate that this did not add any additional value in terms of lowering word error rate.

5. Conclusions

We show that AED models are generally more context sensitive than CTC models and that this difference is largely explained by the attention-mechanism of AED models. Adding a self-attention layer to the CTC model bridges the gap between the models. Analyzing performance by constraining temporal context, we also find that the initial difference between the two models is not crucial in terms of word error rate performance, although both models rely heavily on context for optimal performance. Our experiments on WSJ and LibriSpeech show that CTC models are capable of delivering state-of-the-art results on par with AED models without an external language model. Because of its simplicity and more stable training, CTC is our preferred E2E ASR framework.
6. References


