

A. Appendix

A.1. Dataset statistics

The statistics of the datasets used in this paper is given in Table 3.

Name	Type	Vocab	#Utter.	#Words
LRW	word	500	-	489K
MV-LRS(w) *	word	480	-	1,9M
MV-LRS *	sent.	30K	430K	5M
LRS2	sent.	41K	142K	2M
T1	text	41K	142K	2M
T2 *	text	60K	8M	26M

Table 3: Description of the datasets used for training and testing. We formed MV-LRS(w) by isolating individual word excerpts of the 480 most frequent words, all of which have a count of at least 1000 samples. The statistics for the MV-LRS and the LRS2 datasets include the noisy “pre-train” sets in addition to the main dataset. T1 consists of the transcriptions of the samples in LRS2. We form T2 by collecting the full transcripts of all the subtitles of the shows used in the making of LRS2. The sets marked with * are not publicly available.

A.2. Visual front-end architecture

The details of the spatio-temporal front-end are given in Table 4.

Layer Type	Filters	Output dimensions
Conv 3D	$5 \times 7 \times 7, 64, / [1, 2, 2]$	$T \times \frac{H}{2} \times \frac{W}{2} \times 64$
Max Pool 3D	$/ [1, 2, 2]$	$T \times \frac{H}{4} \times \frac{W}{4} \times 64$
Residual Conv 2D	$[3 \times 3, 64] \times 2 / 1$	$T \times \frac{H}{4} \times \frac{W}{4} \times 64$
Residual Conv 2D	$[3 \times 3, 64] \times 2 / 1$	$T \times \frac{H}{4} \times \frac{W}{4} \times 64$
Residual Conv 2D	$[3 \times 3, 128] \times 2 / 2$	$T \times \frac{H}{8} \times \frac{W}{8} \times 128$
Residual Conv 2D	$[3 \times 3, 128] \times 2 / 1$	$T \times \frac{H}{8} \times \frac{W}{8} \times 128$
Residual Conv 2D	$[3 \times 3, 256] \times 2 / 2$	$T \times \frac{H}{16} \times \frac{W}{16} \times 256$
Residual Conv 2D	$[3 \times 3, 256] \times 2 / 1$	$T \times \frac{H}{16} \times \frac{W}{16} \times 256$
Residual Conv 2D	$[3 \times 3, 512] \times 2 / 2$	$T \times \frac{H}{32} \times \frac{W}{32} \times 512$
Residual Conv 2D	$[3 \times 3, 512] \times 2 / 1$	$T \times \frac{H}{32} \times \frac{W}{32} \times 512$

Table 4: Architecture details for the spatio-temporal visual front-end [16]. The strides for the residual 2D convolutional blocks apply to the first layer of the block only (i.e. the total down-sampling factor in the network is 32). A short cut connection is added after every pair of 2D convolutions [3]. The 2D convolutions are applied separately on every time-frame.

A.3. Seq2Seq decoding with external language model

For decoding with the TM model, we use a left-to right beam search with width W as in [37, 43], with the hypotheses y being scored as follows:

$$\text{score}(x, y) = \frac{\log p(y|x) + \alpha \log p_{LM}(y)}{LP(y)}$$

where $p(y|x)$ and $p_{LM}(y)$ are the probabilities obtained from the visual and language models respectively and LP is a length normalization factor $LP(y) = \left(\frac{5+|y|}{6}\right)^\beta$ [43]. We did not experiment with a coverage penalty. The best values for the hyperparameters were determined via grid search on the validation set: for decoding without the external language model (T1) they

were set to $W = 5$, $\alpha = 0.0$, $\beta = 0.6$ and for decoding with the LM (T2) to $W = 15$, $\alpha = 0.1$, $\beta = 0.7$.

A.4. CTC decoding algorithm with external language model

Algorithm 1 describes the CTC decoding procedure with an external language model. It is also a beam search with width W and hyperparameters α and β that control the relative weight given to the LM and the length penalty. The beam search is similar to the one described for seq2seq above, with some additional bookkeeping required to handle the emission of repeated and blank characters and normalization $LP(y) = |y|^\beta$. We obtain the best results with $W = 100$, $\alpha = 0.5$, $\beta = 0.1$.

Algorithm 1 CTC Beam search decoding with Language Model adapted from [36]. Notation: A is the alphabet; $p_b(s, t)$ and $p_{nb}(s, t)$ are the probabilities of partial output transcription s resulting from paths ending in blank and non-blank token respectively, given the input sequence up to time t ; $p(s, t) = p_b(s, t) + p_{nb}(s, t)$.

Parameters CTC probabilities $p_{1:T}^{ctc}$, word dictionary, beam width W , hyperparameters α, β
initialize $B_t \leftarrow \{\emptyset\}$; $p_b(\emptyset, 0) \leftarrow 1$; $p_{nb}(\emptyset, 0) \leftarrow 0$
for $t = 1$ **to** T **do**
 $B_{t-1} \leftarrow W$ prefixes with highest $\frac{\log p(s, t)}{|s|^\beta}$ in B_t
 $B_t \leftarrow \{\}$
 for prefix s in B_{t-1} **do**
 $c^- \leftarrow$ last character of s
 $p_b(s, t) \leftarrow p_t^{ctc}(-, t)p(s, t-1)$ ▷ adding a blank
 $p_{nb}(s, t) \leftarrow p_t^{ctc}(c^-, t)p_{nb}(s, t-1)$ ▷ repeated
 add s to B
 for character c in A **do**
 $s^+ \leftarrow s + c$
 if s ends in c **then**
 $p_c \leftarrow p_t^{ctc}(c, t)p(c, t-1)p_{LM}(c|s)^\alpha$
 else
 ▷ repeated chars must have blanks in between
 $p_c \leftarrow p_t^{ctc}(c, t)p_b(c, t-1)p_{LM}(c|s)^\alpha$
 if s^+ is already in B_t **then**
 $p_{nb}(s^+, t) \leftarrow p_{nb}(s^+, t) + p_c$
 else
 add s^+ to B_t
 $p_{nb}(s, t) \leftarrow 0$
 $p_{nb}(s^+, t) \leftarrow p_c$
 return $\max_{s \in B_t} \frac{\log p(s, T)}{|s|^\beta}$ in B_T

A.5. Online CTC decoding algorithm

Algorithm 2 describes the online CTC decoding procedure introduced in Section 4.

Algorithm 2 Online CTC decoding with fully convolutional model. The algorithm runs in $O(TrW|A|)$ time, where T denotes the input sequence length, r is half the length of the network’s total receptive field, W the beam width and $|A|$ the number of characters in the alphabet. The BeamStep routine performs one step of the CTC Beam Search decoding outer loop shown in Algorithm 1

Parameters Input video frames $x_{1:T}$, FC network f_θ
 initialize $\mathbf{B}_0 \leftarrow \{\emptyset\}$; $\mathbf{L}_0 \leftarrow \{\emptyset\}$; \triangleright Beam & LM states
for $t = 1$ **to** T **do** \triangleright decoding steps lag by r behind real time
 $p_{t:t+r}^{ctc} \leftarrow f_\theta(x_{t-r:t})$ \triangleright Slide network right by one step
 $\mathbf{B}_t, \mathbf{L}_t \leftarrow \text{BEAMSTEP}(p_{t:t+r}^{ctc}, \mathbf{B}_t, \mathbf{L}_t)$ $\triangleright O(W \cdot |A|)$
 $\hat{\mathbf{B}}_t, \hat{\mathbf{L}}_t \leftarrow \text{copy } \mathbf{B}_t, \mathbf{L}_t$
 for $\tau = t + 1$ **to** $t + r$ **do**
 $\hat{\mathbf{B}}_\tau, \hat{\mathbf{L}}_\tau \leftarrow \text{BEAMSTEP}(p_\tau^{ctc}, \hat{\mathbf{B}}_\tau, \hat{\mathbf{L}}_\tau)$ $\triangleright O(W \cdot |A|)$
 $D_t \leftarrow$ highest scoring sentence $S \in \hat{\mathbf{B}}_{t+r}$
return D_T

A.6. Confusion Matrix

Figure 2 shows the confusion between the predictions of the FC-15 model obtained with greedy decoding of the CTC posteriors.

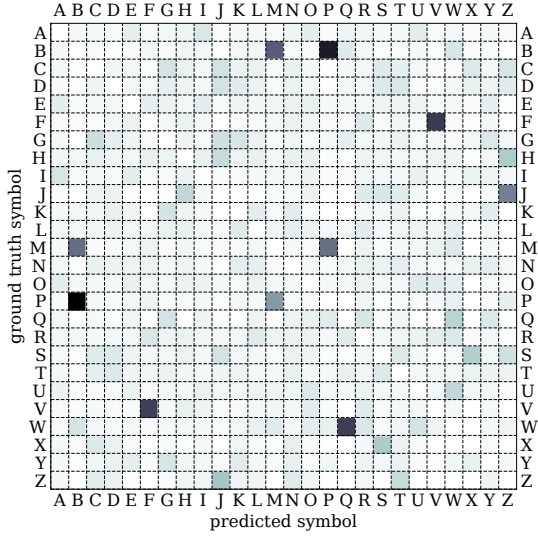


Figure 2: Confusion matrix of CTC predictions. The entries of the table are the normalized substitution counts obtained from the minimum edit distance calculation between ground truth and sentences predicted with greedy CTC decoding, averaged over the whole dataset. We observe that the network confuses characters that are difficult to discriminate between using the visual information alone. For example B is frequently confused with M and P, while V is confused with F and vice versa. It is interesting to note that Q is often emitted instead of W. We hypothesize that this happens because these characters appear very similar visually in words like ‘week / quick’, ‘woe / quote’.

A.7. Decoding examples

Table 5 shows further examples of online decoding outputs.

frame #	Decoded string	Decoded string starting from the middle (frame #55)
002	one	
007	to	
010	it in	
011	on it	
012	to on	
013	to how	
014	at home	
026	at home	
027	at home and	
028	home	
029	home to	
032	home to	
033	home to your	
038	home you	
040	home you are	
041	home you and	
045	home to you and	
046	home to you and had	
047	home you and had	
048	home you and adam	
051	home you and animals	
054	home to an animal	
056	home you and animals	i
058		in
059		it in
060	home to an animal	i and
061	home to an animal and	in
062		in the
063	home to an animal that	then the
064		that is
066	home to an animal that	that here
067	home to an animal that is	
068		that is
070		that it's
070		that is
074	home to an animal that is	that it's
075	home to an animal that is right	that it's right
076	home to an animal that it's right	
078	home to an animal that is right	
081	home to an animal that is right in	that is right in
082		that it's right in
083	home to an animal that is right in the	that it's right in the
087	home to an animal that is right in the training	that it's right in the town
089	home to an animal that is right in the town	
090	home to an animal that it's right in the top	that it's right in the top
091	home to an animal that it's right in the top	that it's right in the top
092	home to an animal that it's right in the top of	that it's right in the top of
094	home to an animal that it's right in the top of	that it's right in the top of
097	home to an animal that it's right in the top of the	that it's right in the top of the
098	home to an animal that it's right in the top of the room	that it's right in the top of the front
099		that it's right in the top of the room
100	home to an animal that it's right in the top of the food	that it's right in the top of the foot
101		that it's right in the top of the food
102	home to an animal that it's right in the top of the foot	that it's right in the top of the foot
103	home to an animal that it's right in the top of the future	that it's right in the top of the future
# changes/frame	0.4	0.5

Table 5: Example of sequential online decoding starting the beginning (*left*) and from the middle of the utterance (*right*). The ground truth transcription is “home to an animal that is right at the top of the food chain”. Red color denotes the completions of words by the language model. It can be seen that after some initial frames where the model does not have enough context to make a confident prediction, it starts predicting correctly.