

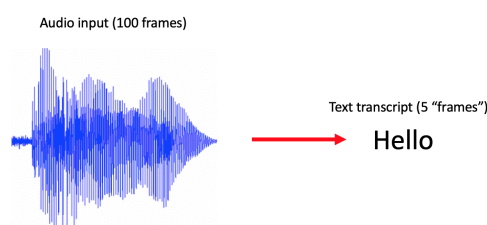
# A Fully Differentiable Beam Search Decoder

## Notes by Bun

### 1. Motivation

This paper addresses the problems in sequence modelling on unaligned sequences. More formally, we would like to learn the mapping between  $X = [x_1, x_2, \dots, x_T]$  and  $Y = [y_1, y_2, \dots, y_U]$  ( $T \geq U$ ).

Where does this kind of problem appear? Consider speech recognition. We have a dataset of audio clips and their corresponding transcripts. Unfortunately, we don't know how the characters in the transcript align to the audio. This makes training a speech recognizer harder than it might at first seem.



There are currently two general approaches to dealing with this alignment issue, Connectionist Temporal Classification (CTC) or Seq2Seq (with attention) modelling.

CTC learns a model that outputs a distribution over labels/characters for every for input frame,  $Z = [z_1, z_2, \dots, z_T]$  (outputs are independent, subsequent outputs on conditioned on previous outputs), creating an outputs of length  $T$ . By marginalising over all valid outputs that could generate the target, CTC can learn  $P(Y|X)$ . The loss is then the negative log likelihood.

$$p(Y|X) = \sum_{Z \in \text{valid sequences}} \prod_{t=1}^T p(z_t|X) \quad (1)$$

Outputs that could produce 'hello'  
for audio sample of 10 frames

Valid outputs	Invalid outputs
'hheelllloo'	'hehellloo'
'hhhhelloo'	'hheloloo'
'hellooooo'	'hhaellloo'
'heeeelloo'	'helllooloo'

Seq2seq instead learns an implicit alignment between the the input and target which is then optimised via cross-entropy. Unlike CTC, in seq2seq learning, outputs are not conditionally independent.

At inference, both models try to find the mode of their distributions

$$Y^* = \operatorname{argmax} P(Y|X) \quad (2)$$

As enumerating all possible sequences is intractable, beam search is a popular choice for reducing the hypothesis space. Occasionally, hypotheses will be post-processed by a language model to generate a high probability  $Y$ .

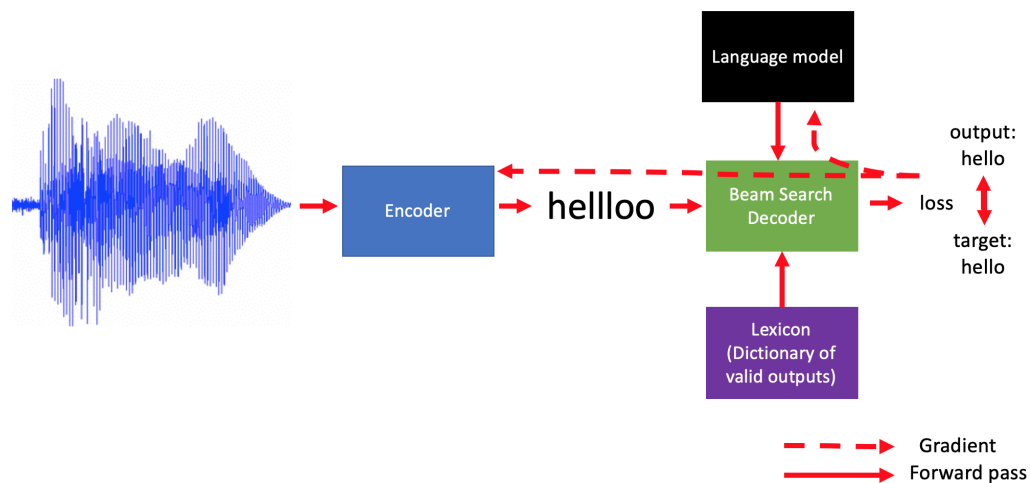
Both these frameworks fall victim to two issues.

1. Exposure bias (training objective and test objectives are different)
2. Label bias (Token level normalisation are not equal to sequence level normalisation, probabilities of previous tokens cannot be "revised" after future tokens are selected)

This work aims to solve both these issues by:

1. Utilising a differentiable beam-search decoder during training
2. Using unnormalised token level scores that are normalised on a sequence level by a language model

As the decoder is differentiable both the encoder (that outputs  $Z$ ) and LM can be trained jointly.



## 2. Model

Similar to CTC, the model aims to estimate

$$\log p(Y|X) = \log \sum_{\substack{Z \text{ can be} \\ \text{merged into } Y}} p(Z|X) \quad (3)$$

However, instead of using normalised token level scores  $p(z_t|X)$ , un-normalised token scores are utilised.

$$s(Z|X) = h(Y) + \sum_{t=1}^T f(z_t|X) + g(z_t|z_{t-1}) \quad (4)$$

Note in this case, the token transition model,  $g$ , is considered to be a bigram for simplicity (larger n-gram models will require exponentially increasing number of terms for renormalisation later). A sequence level normalisation is included,  $h$ , through a (character-level) language model.

Renormalising this term requires summation over all possible sequences.

$$\log p(z_t|X) = s(Z|X) - \sum_{V \in \text{all sequences}} \exp s(V|X) \quad (5)$$

Combining equation (3) and (5)

$$\log p(Y|X) = \logadd_{\substack{Z \text{ can be} \\ \text{merged into } Y}} s(Z|X) - \logadd_{V \in \text{all sequences}} s(V|X) \quad (6)$$

where  $\logadd(a, b) = \log(e^a + e^b)$

These summations are obviously highly intractable for large number of input frames. To address this some approximations have to be made.

### 2.1 Approximation

Using a lexicon/dictionary (implemented as a trie), we can constrain the first sum only include sequences composed of valid sequences of words in the dictionary. The set of sequences considered by this sum will be denoted,  $W$ .

To constrain the normalising term (the second sum), instead of considering all sequences, consider a beam containing the top-k hypotheses,  $K$ , that maximise  $s(Z|X)$ . We can perform normalisation only using the hypotheses in this beam.

$$\log p(Y|X) = \logadd_W s(Z|X) - \logadd_K s(V|X) \quad (7)$$

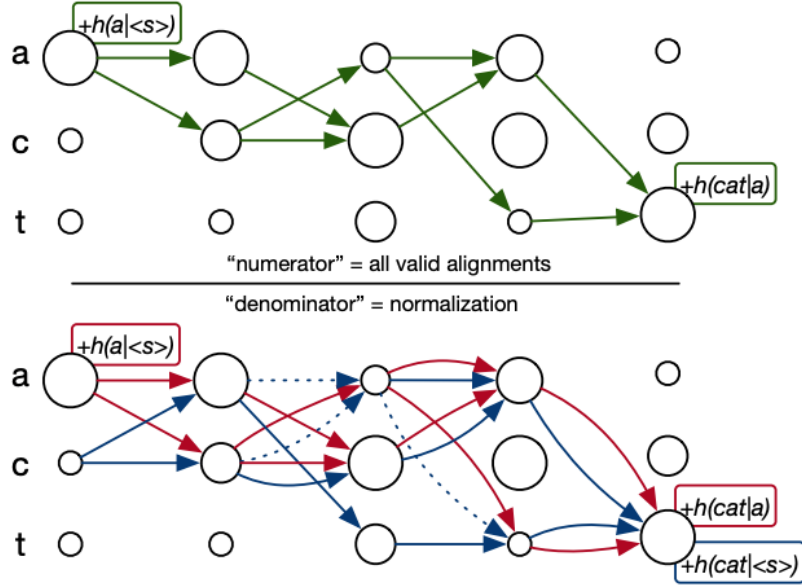


Figure 1: An example of the DBD computation of the loss (Equation (7)), with a target transcription of “a cat”, using a lexicon {a, cat}, 5 frames in total, and a word-level bigram LM  $h(\cdot)$ . Circle sizes are proportional to the AM score and paths through the graph are aggregated with a logadd. The first term (maximized, “numerator”) corresponds to all valid alignments (in green). The second term corresponds to the beam search which is used to construct the denominator (Equation (8)). Dashed arrows denote transitions not included in the beam. Multiple alignments leading to the same word are merged, and the LM scores are added (rounded rectangles) as words are considered in the beam.

Note:  $K$  may not contain all the sequences in  $W$ . As such, the normalisation term set is actually  $K^* = K \cup W$

### 3. Results

Table 1: Comparing WER performance of ASG with decoding grid-search, and DBD, on WSJ. We compare with standard end-to-end approaches, for reference.

Model	nov93dev	nov92
ASG 10M AM (beam size 8000)	8.5	5.6
ASG 10M AM (beam size 500)	8.9	5.7
ASG 7.5M AM (beam size 8000)	8.8	6.0
ASG 7.5M AM (beam size 500)	9.4	6.1
DBD 10M AM (beam size 500)	8.7	5.9
DBD 7.5M AM (beam size 500)	7.7	5.3
DBD 7.5M AM (beam size 1000)	7.7	5.1
Attention RNN+CTC (3gram) (Bahdanau et al., 2016a)		9.3
CNN+ASG (4-gram) (Zeghidour et al., 2018)	9.5	5.6
CNN+ASG (wav+convLM) (Zeghidour et al., 2018)	6.8	3.5
RNN+E2E-LF-MMI (data augm.) (+RNN-LM) (Hadian et al., 2018)		4.1
BLSTM+PAPB+CE (RNN-LM) (Baskar et al., 2018)		3.8

Table 2: WSJ performance (WER), using only the acoustic training data. ASG n-gram decoding hyper-parameters were tuned via grid-search. Beam size for both ASG and DBD was 500. Larger beam sizes with ASG did not lead to significant improvements.

Model	nov93dev	nov92
ASG ( <i>zero LM decoding</i> )	18.3	13.2
ASG ( <i>2-gram LM decoding</i> )	14.8	11.0
ASG ( <i>4-gram LM decoding</i> )	14.7	11.3
DBD <i>zero LM</i>	16.9	11.6
DBD <i>2-gram LM</i>	14.6	10.4
DBD <i>2-gram-bilinear LM</i>	14.2	10.0
DBD <i>4-gram LM</i>	13.9	9.9
DBD <i>4-gram-bilinear LM</i>	14.0	9.8
RNN+CTC		30.1
(Graves and Jaitly, 2014)		
Attention RNN+CTC		18.6
(Bahdanau et al., 2016a)		
Attention RNN+CTC+TLE		17.6
(Bahdanau et al., 2016b)		
Attn. RNN+seq2seq+CNN		9.6
(speaker adapt.) (Chan et al., 2017)		
BLSTM+PAPB+CE		10.8
(Baskar et al., 2018)		