Метрики качества

WER, CER, SER

какого дьявола ты здесь шумишь

какого дявола ты здесь шумишь

C

$$\text{WER} = \frac{S + D + I}{N}$$

S

where...

S = number of substitutions

D = number of deletions

I = number of insertions

N = number of words in the reference

chars

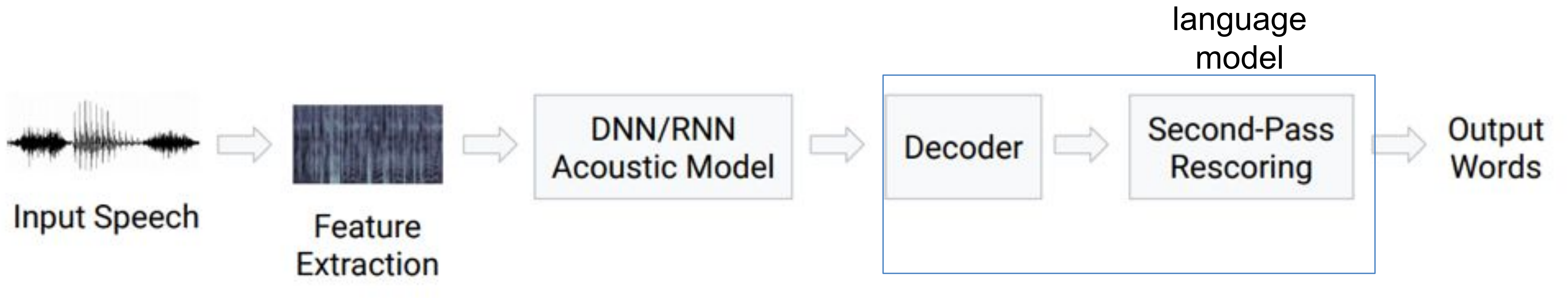
sentences

ASR pipeline

(Hybrid ASR)

Conventional ASR

Pipeline



https://habr.com/ru/company/ru_mts/blog/468663/

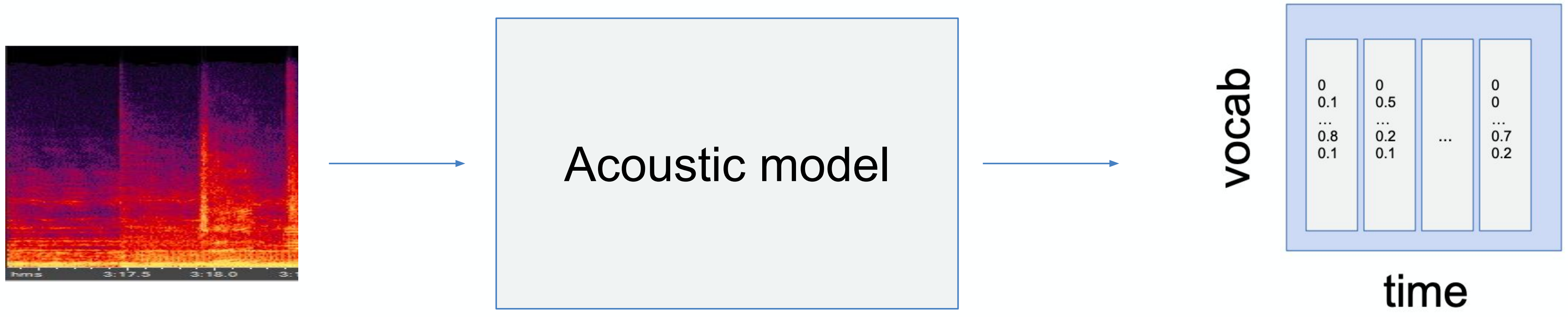
Пайплайн

wav -> melspectrogram (wav2vec) -> Acoustic Model -> Decoding with language model -> (punc) -> words

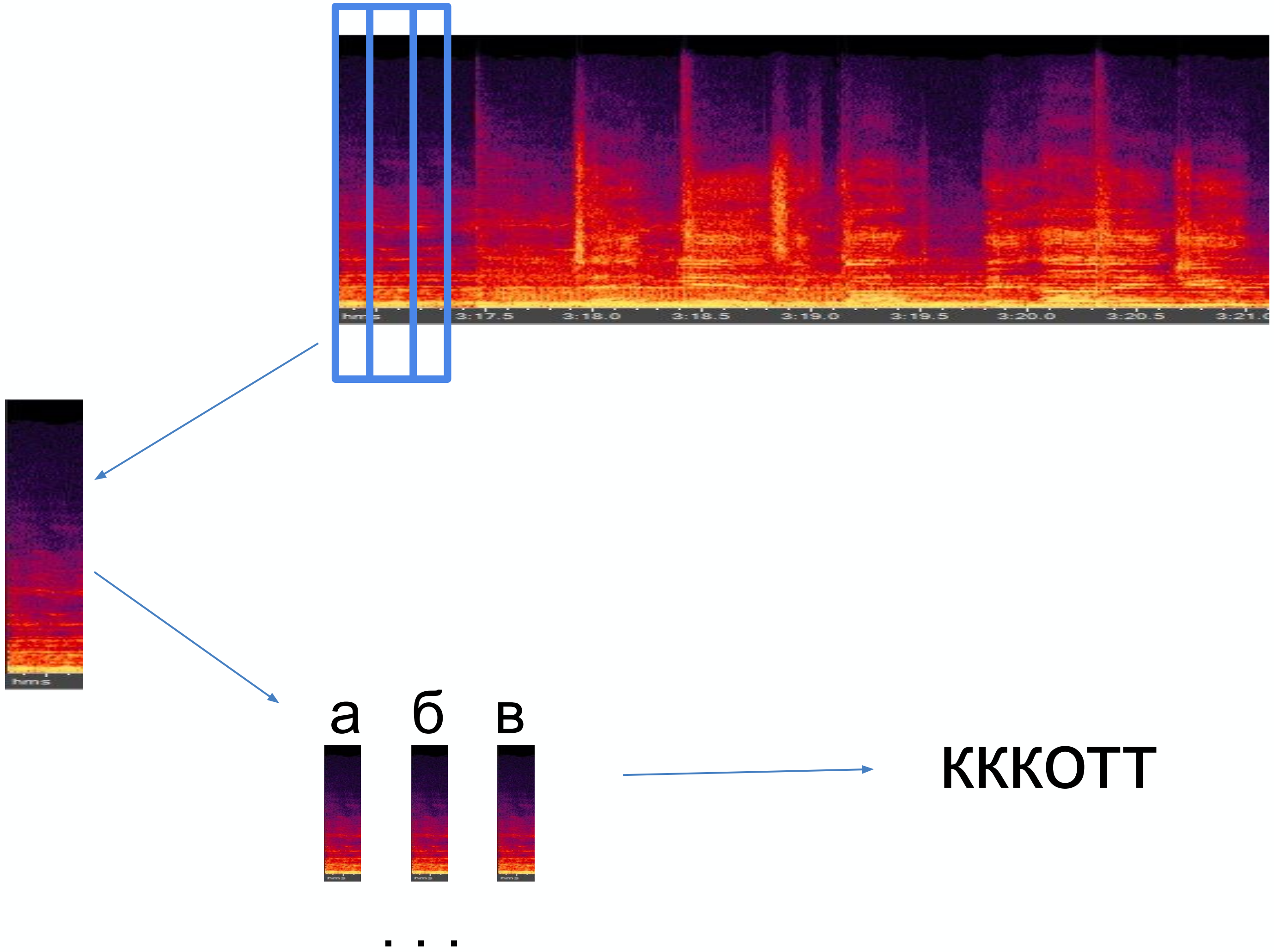
Метрика

wer (word error rate)

Акустическая модель


hms
3:17.5
3:18.0
Acoustic model
vocab
0
0.1
...
0.8
0.1
0
0.5
...
0.2
0.1
...
0
0
...
0.7
0.2
time

frame-level prediction?



CTC-loss

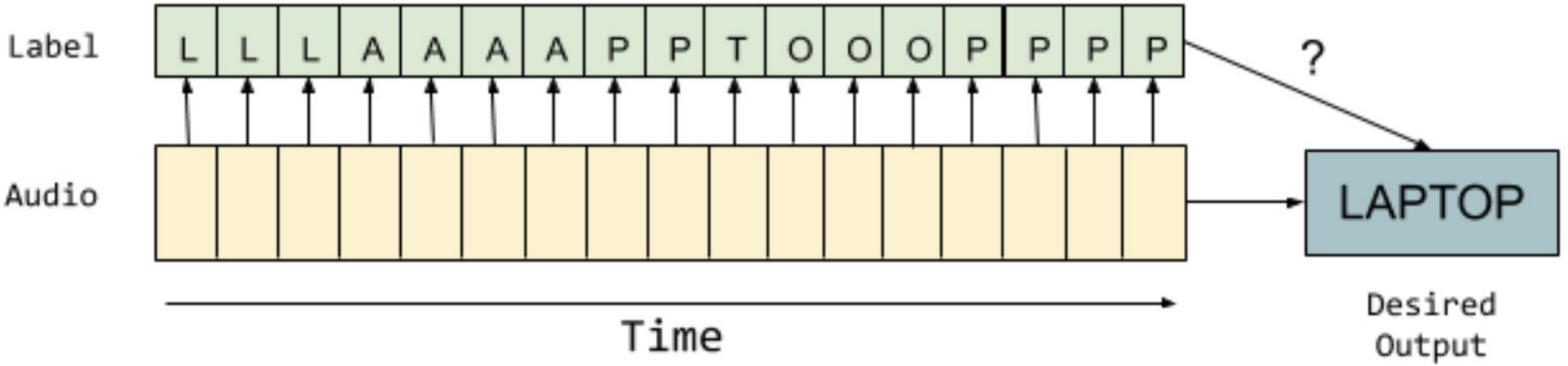


Figure 1. Here, we have aligned audio data, where the audio is chopped up into time slices and each is labeled with a letter. But it's very difficult to go from those labels to the correct transcript, especially considering words with repeated letters (such as "book").

log (Pr (output: "BOOK" | audio)) = log (Pr (BOO-OOO - KK | audio)) + log (Pr (BBO - OO-KKK | audio)) + ...).

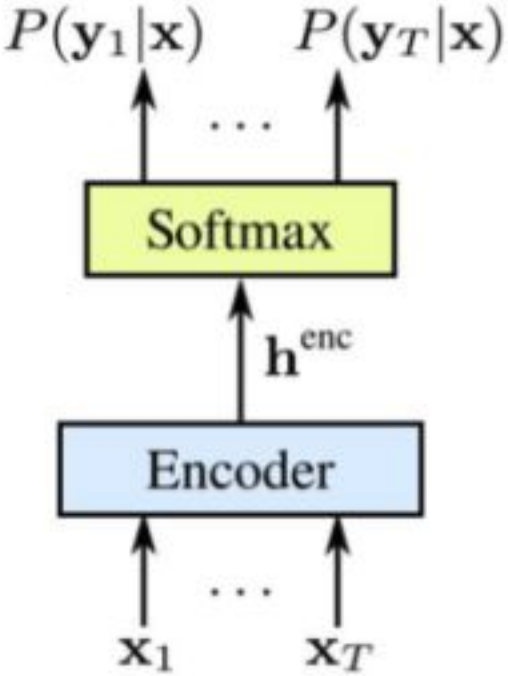
На практике мы можем использовать подход динамического программирования, чтобы рассчитать это, накапливая наши логарифмические вероятности по разным «путям» через выходы softmax на каждом шаге.

коллапсирование

_ _ кккллллаасс_с-с _

_

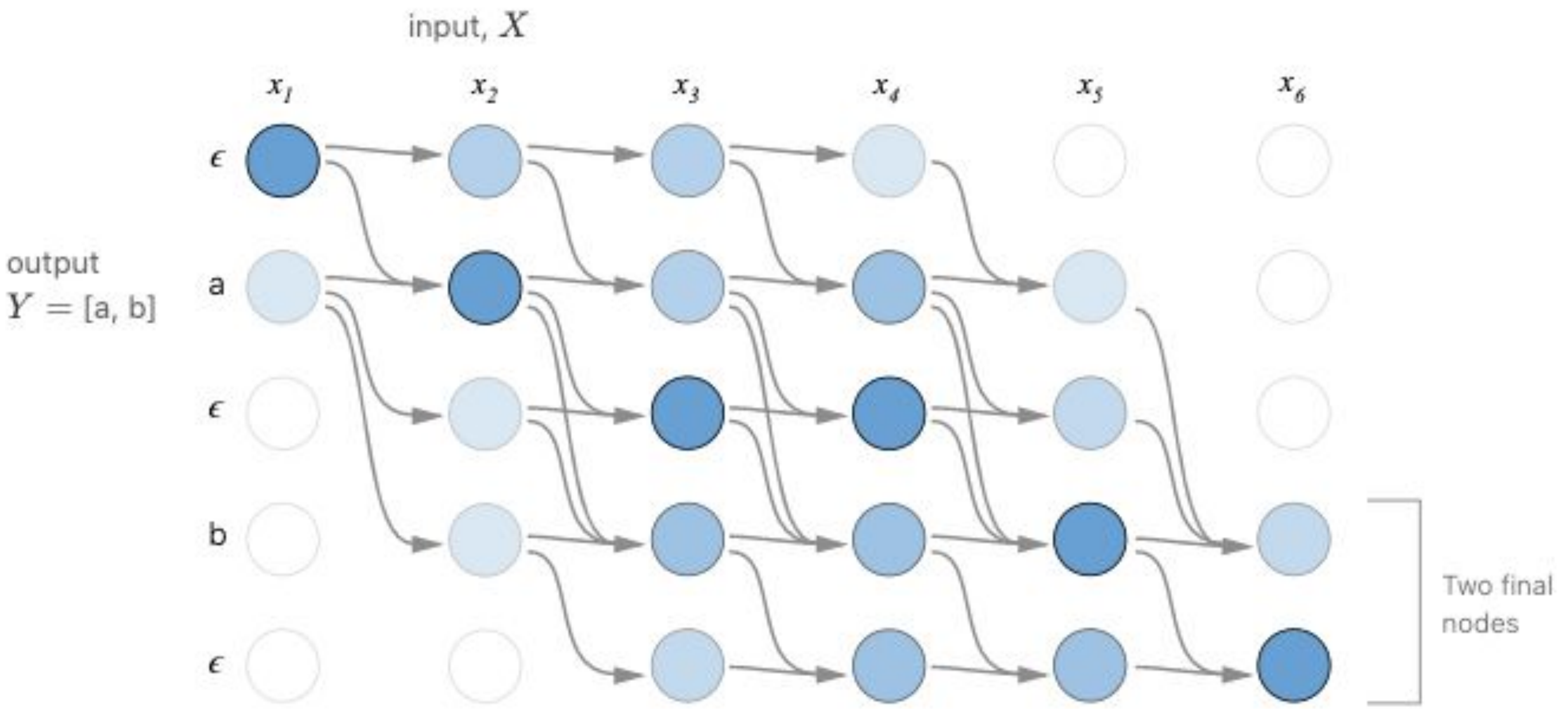
класс



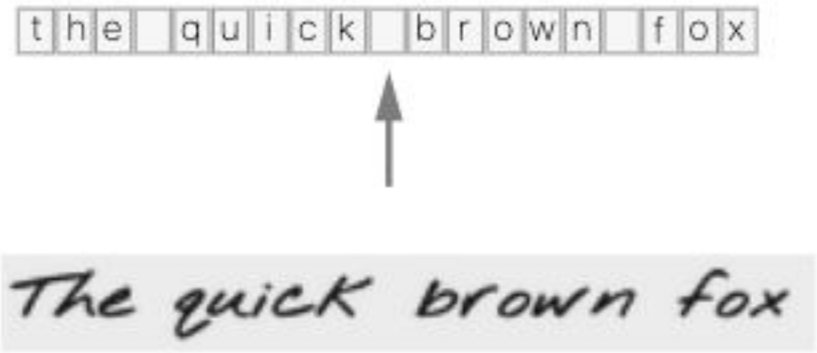
B B **c** B B **a** **a** B B **t**
B **c** **c** B **a** B B B B **t**
...
B **c** B B **a** B B **t** **t** B

$$P(\mathbf{y}|\mathbf{x}) = \sum_{\hat{\mathbf{y}} \in \mathcal{B}(\mathbf{y},\mathbf{x})} \prod_{t=1}^{T} P(\hat{y}_t|\mathbf{x})$$

CTC loss



Node (s, t) in the diagram represents $\alpha_{s,t}$ – the CTC score of the subsequence $Z_{1:s}$ after t input steps.

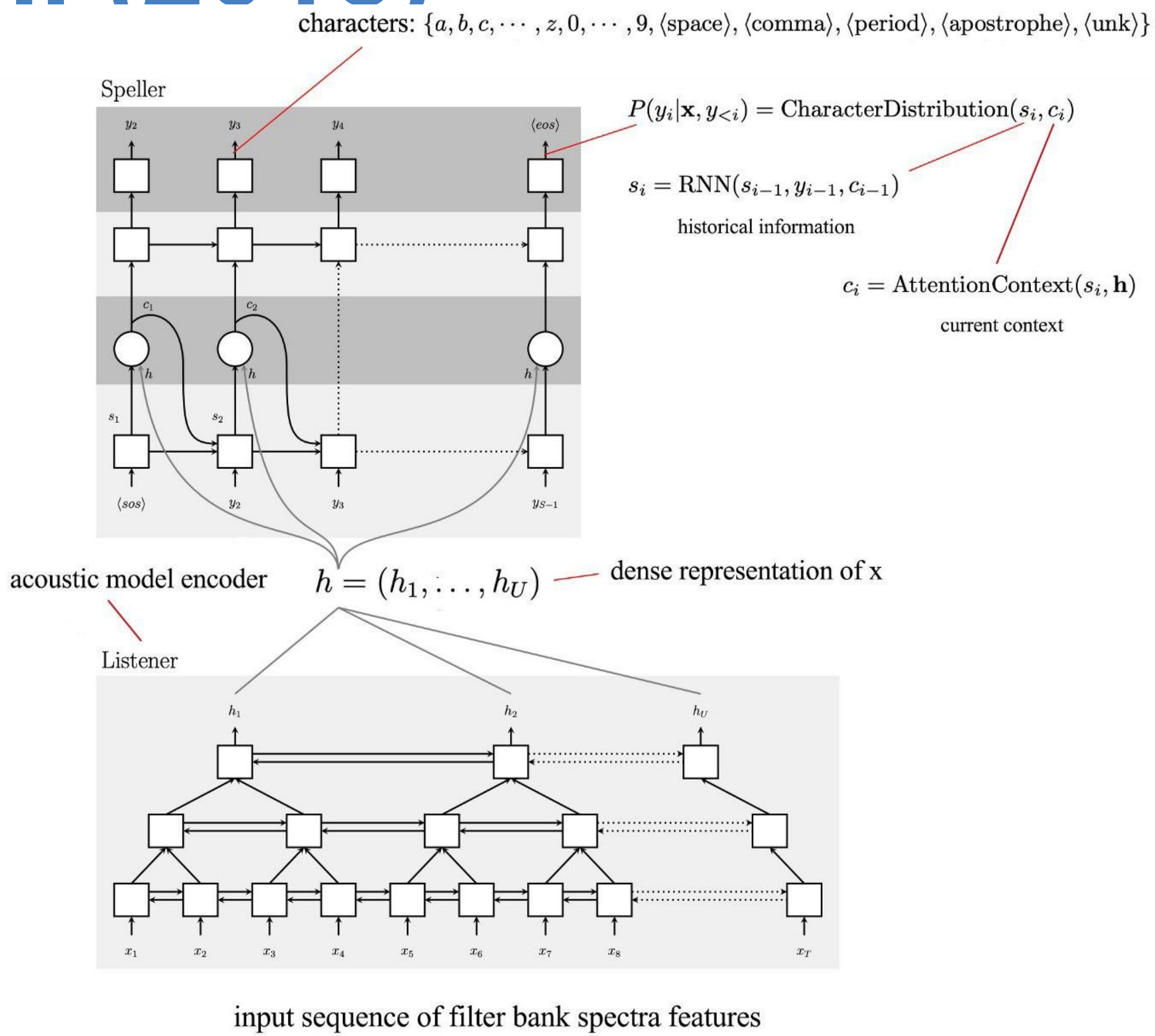


Handwriting recognition: The input can be (x, y) coordinates of a pen stroke or pixels in an image.

Speech recognition: The input can be a spectrogram or some other frequency based feature extractor.

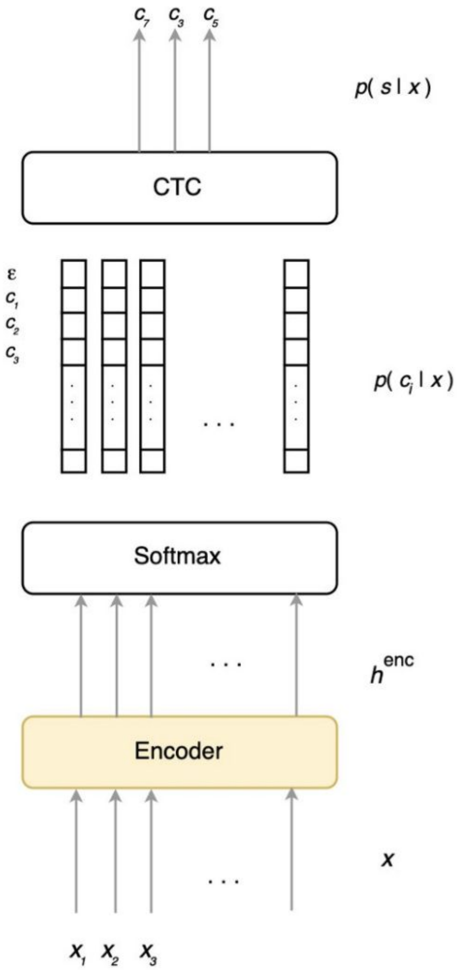
Listen-Attent-Spell (2015)

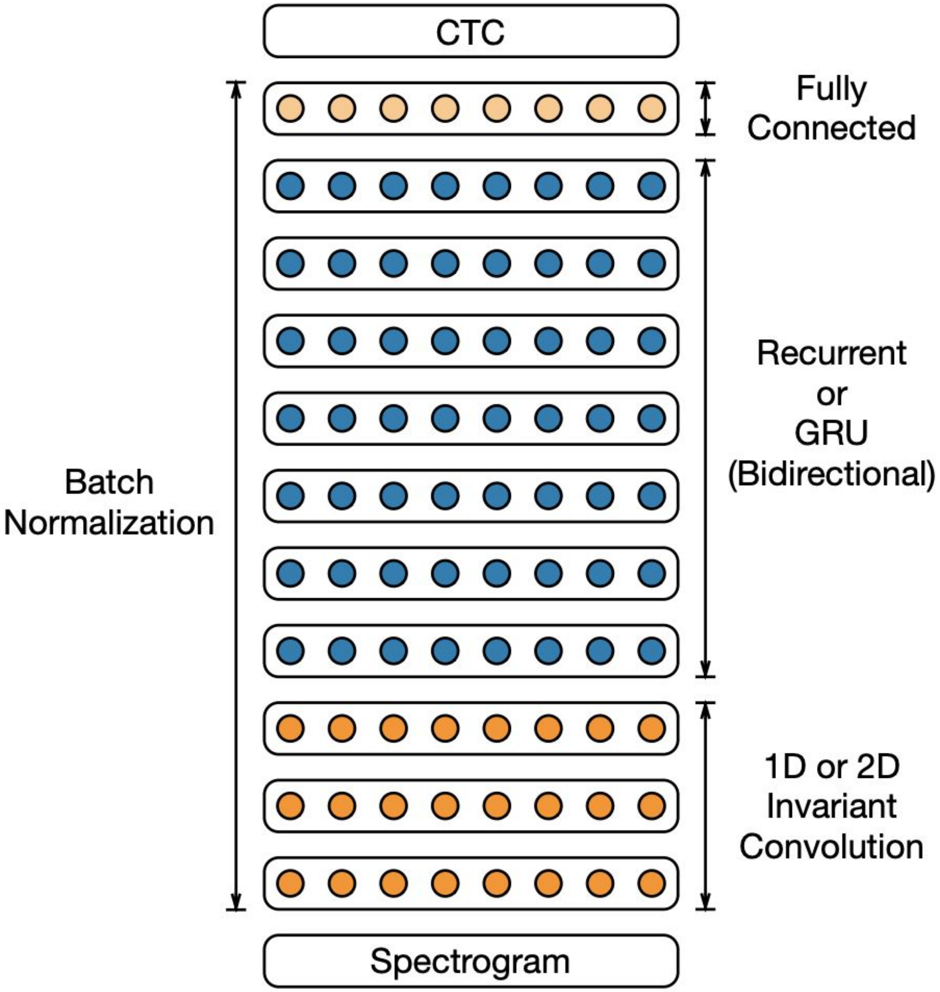
- RNN
- Autoregressive
- No need beam search & LM
- Cross-Entropy



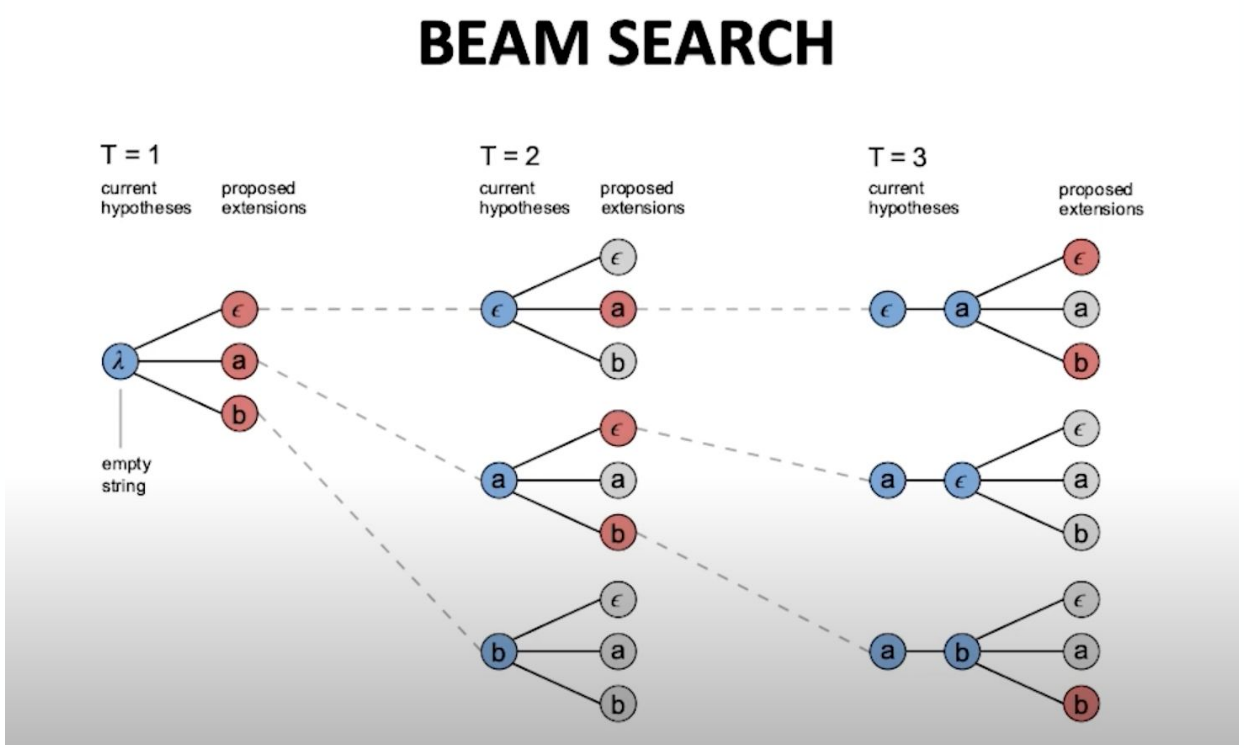
DeepSpeech 2 (2015)

- RNN & Conv
- Non-Autoregressive
- Need LM beam search & LM
- CTC





Beam Search & LM



Only beam search

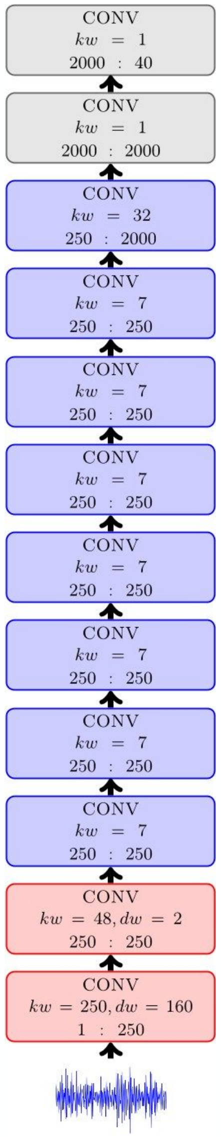
$$y^* = \arg\max_{y} \log p(y|x)$$

Beam search & LM (shallow fusion)

$$y^* = \arg\max_{y} \log p(y|x) + \lambda \log p_{LM}(y)$$

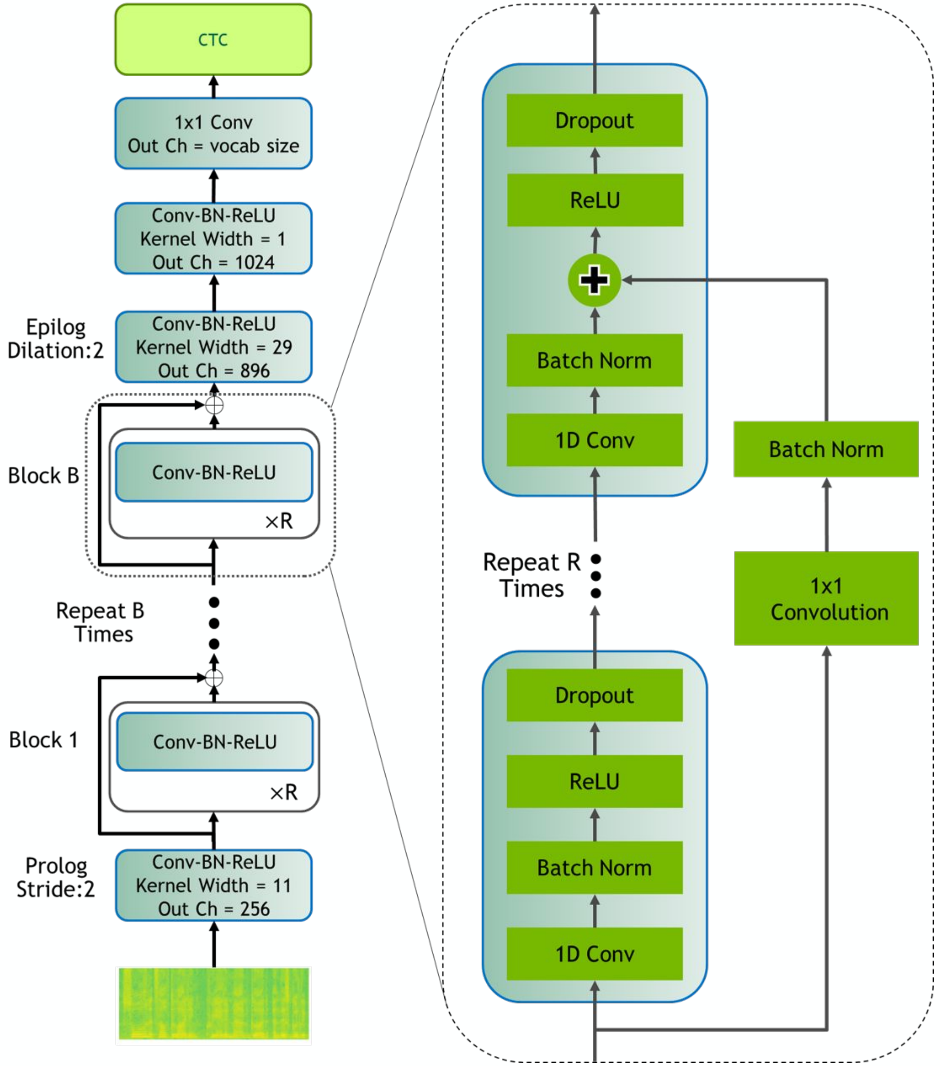
Wav2Letter (2016)

- Conv
- Non-Autoregressive
- Need beam-search & LM
- CTC



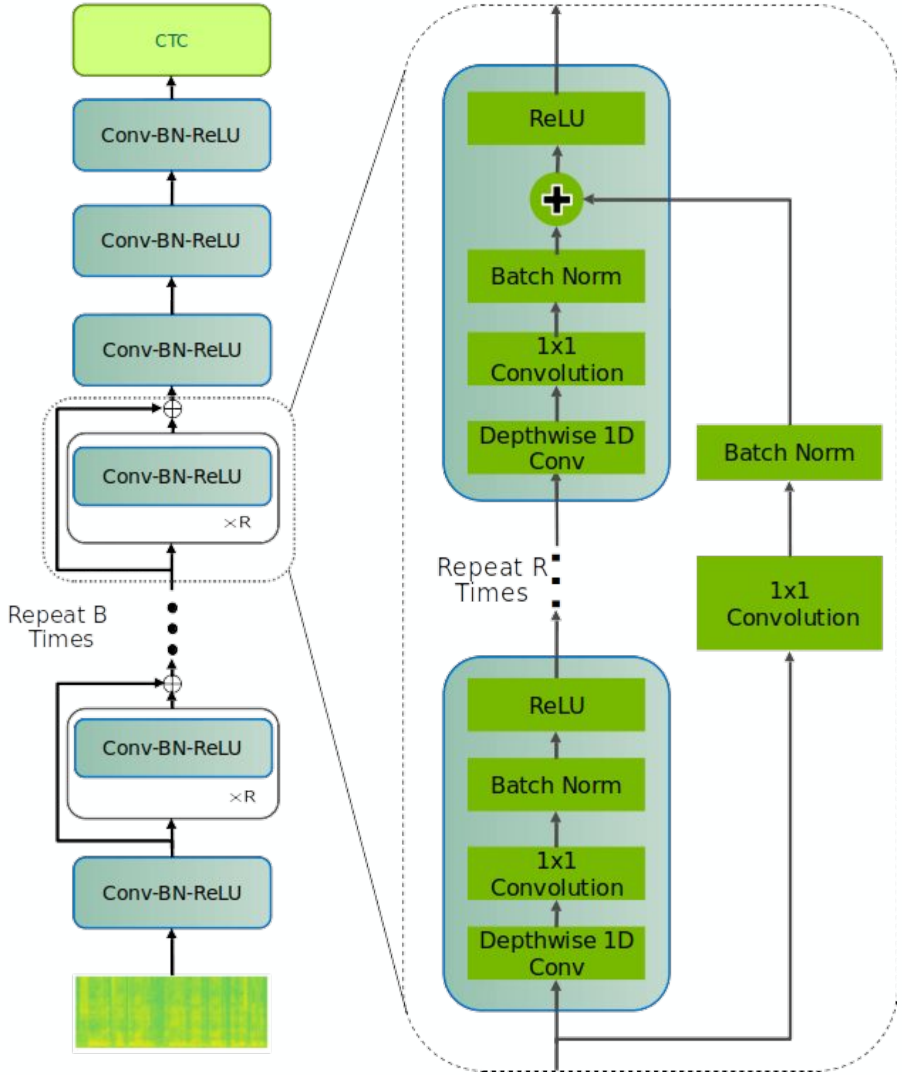
Jasper (2019)

- Conv
- Non-Autoregressive
- Need beam-search & LM
- CTC



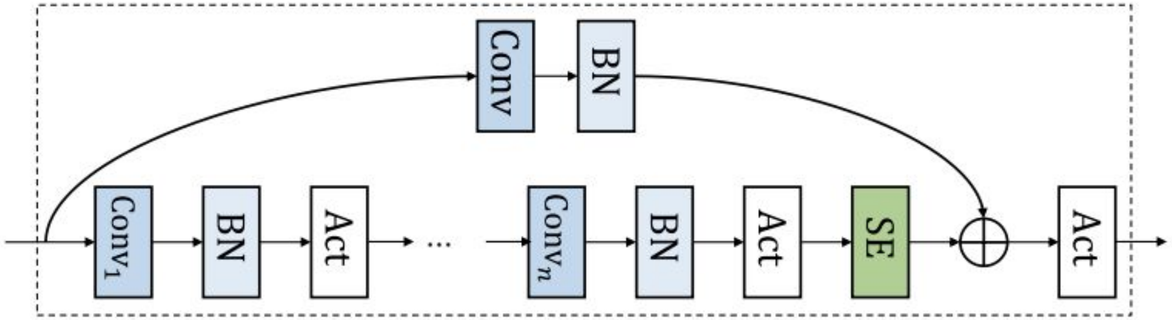
QuartzNet (2019)

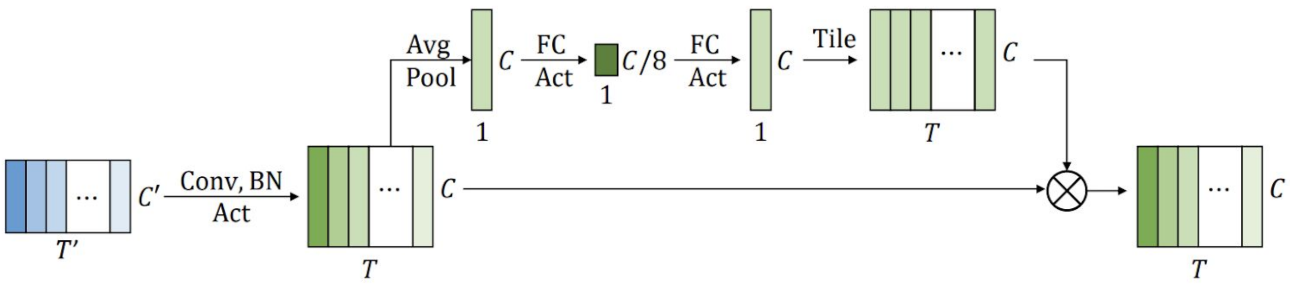
- Conv
- Non-Autoregressive
- Need beam-search & LM
- CTC



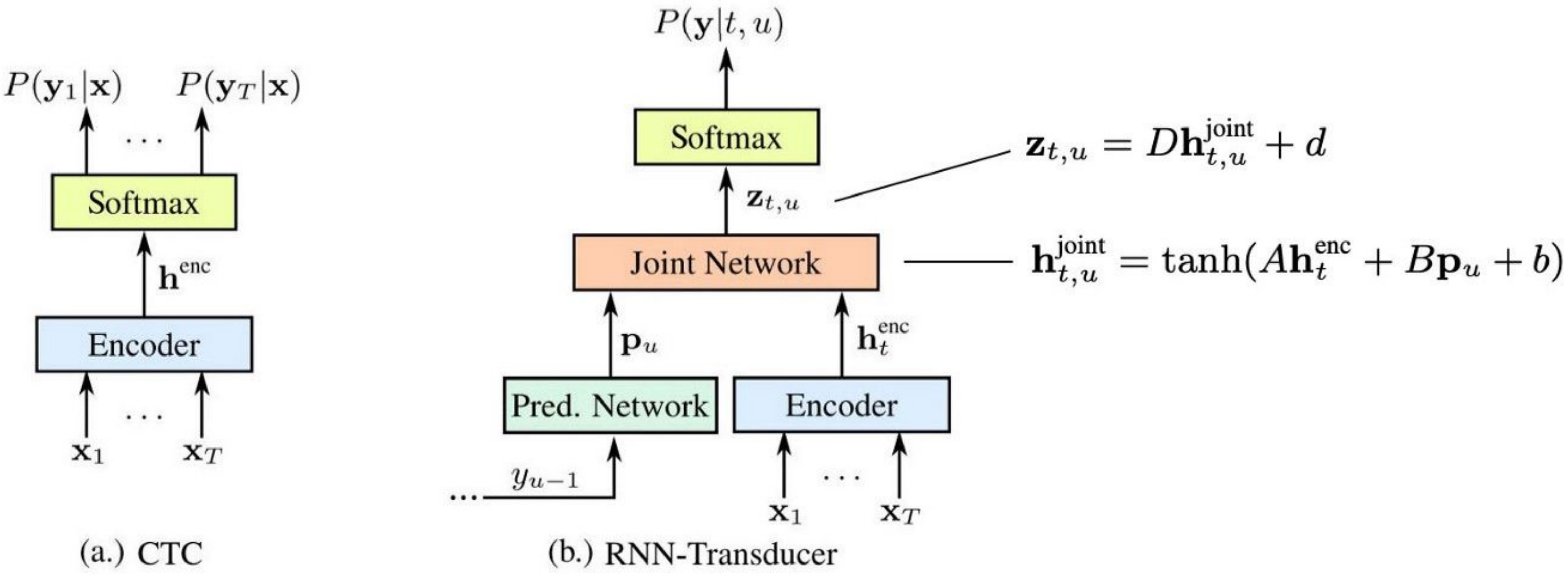
ContextNet (2020)

- RNN & Conv
- Autoregressive
- Better with beam-search & LM, but can work without it
- RNN-T loss





RNN-Transducer



Conformer (2020)

- RNN, Conv & Transformer
- Autoregressive
- Better with beam-search & LM, but can work without it
- RNN-T loss

Method	#Params (M)	WER Without LM		WER With LM	
		testclean	testother	testclean	testother
Hybrid					
Transformer [33]	-	-	-	2.26	4.85
CTC					
QuartzNet [9]	19	3.90	11.28	2.69	7.25
LAS					
Transformer [34]	270	2.89	6.98	2.33	5.17
Transformer [19]	-	2.2	5.6	2.6	5.7
LSTM	360	2.6	6.0	2.2	5.2
Transducer					
Transformer [7]	139	2.4	5.6	2.0	4.6
ContextNet(S) [10]	10.8	2.9	7.0	2.3	5.5
ContextNet(M) [10]	31.4	2.4	5.4	**2.0**	4.5
ContextNet(L) [10]	112.7	**2.1**	4.6	**1.9**	4.1
Conformer (Ours)					
Conformer(S)	10.3	**2.7**	**6.3**	**2.1**	**5.0**
Conformer(M)	30.7	**2.3**	**5.0**	**2.0**	**4.3**
Conformer(L)	118.8	**2.1**	**4.3**	**1.9**	**3.9**