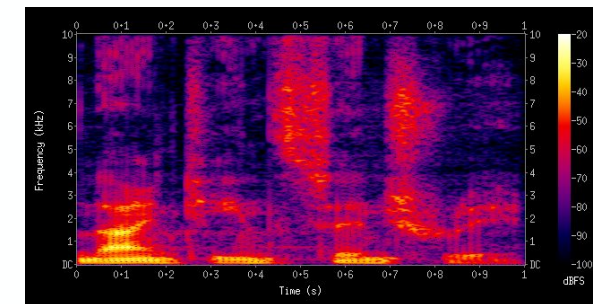
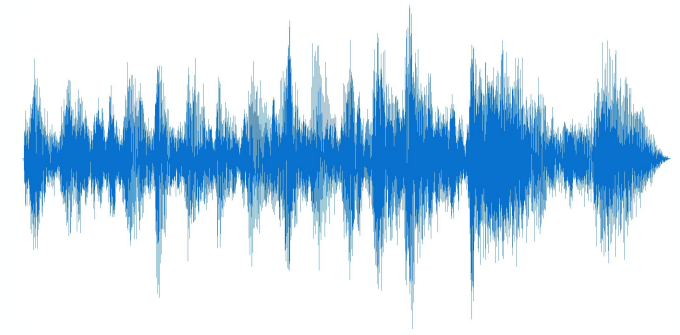
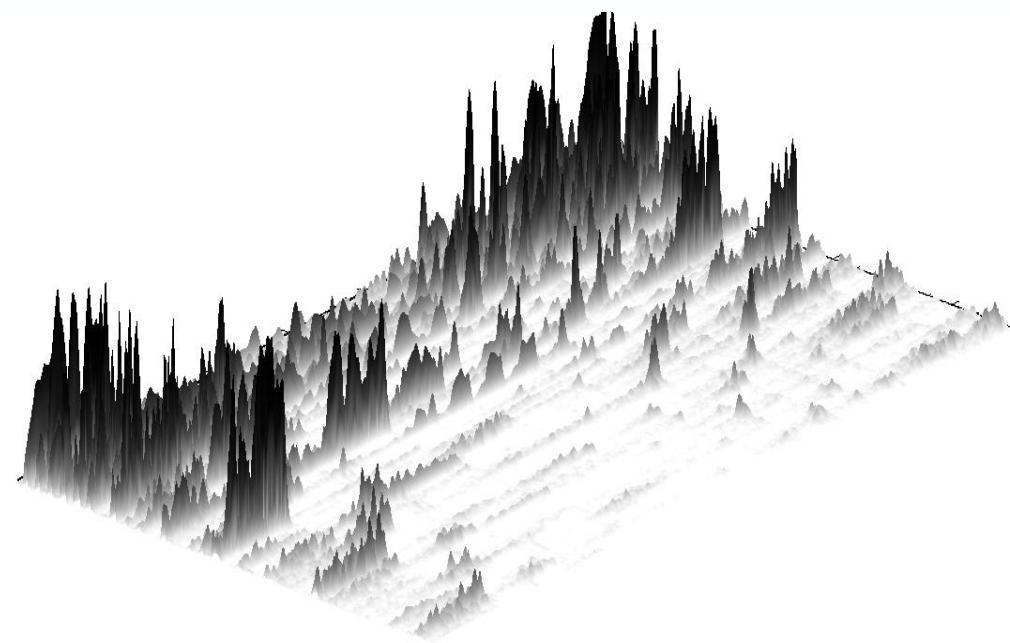
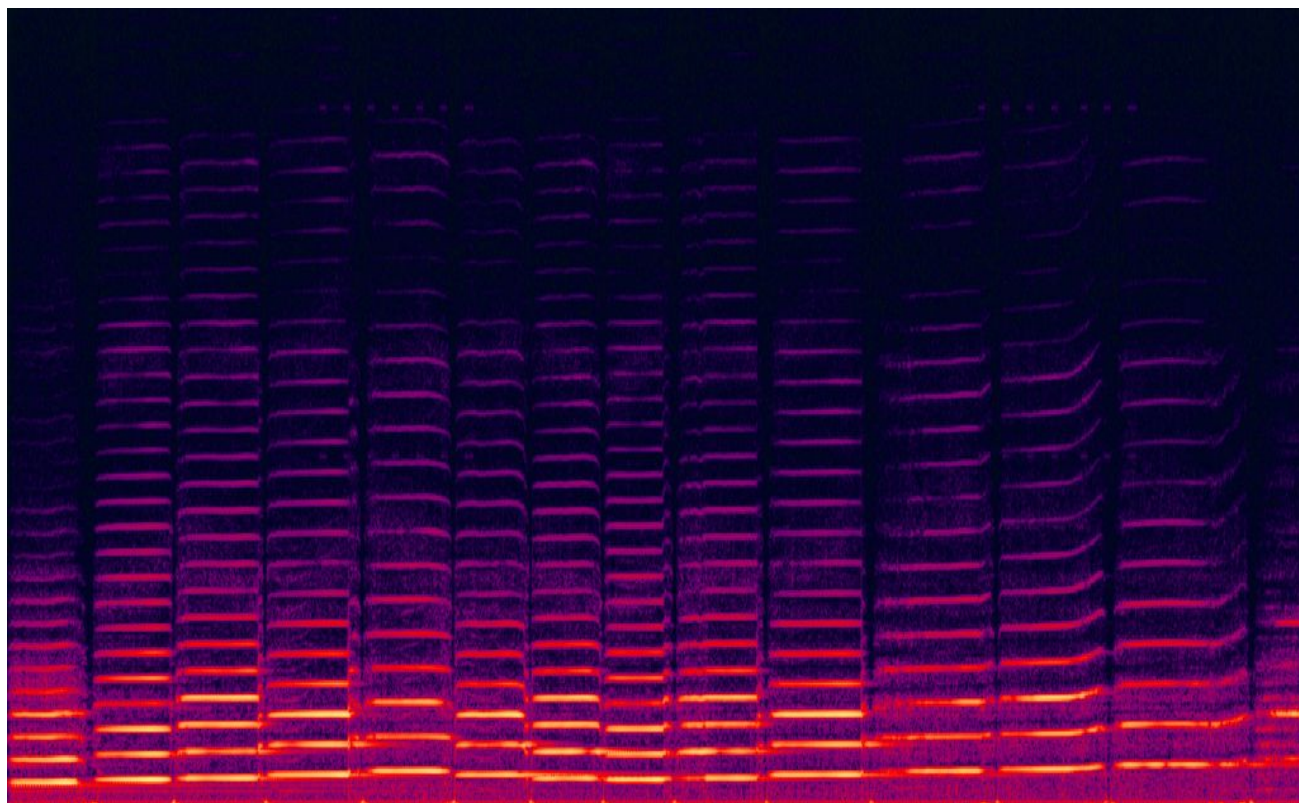
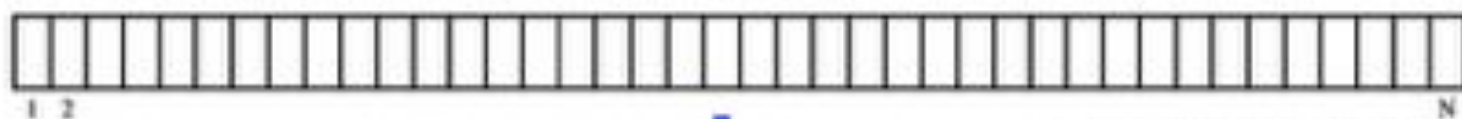


- Препроцессинг
- Акустическая модель
- Языковая модель
- Пунктуационная модель



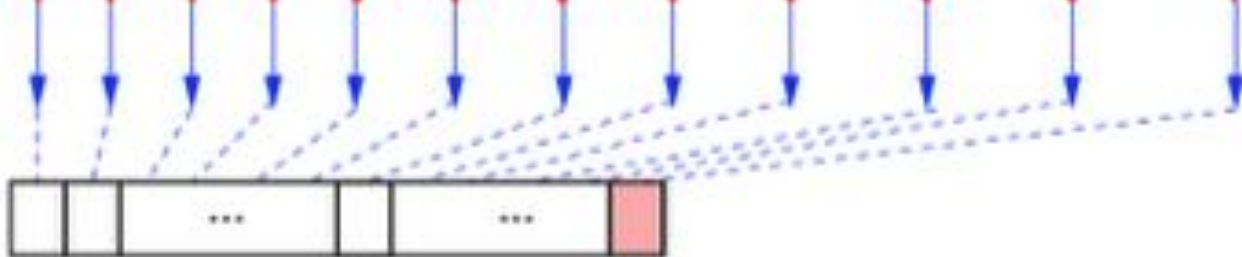
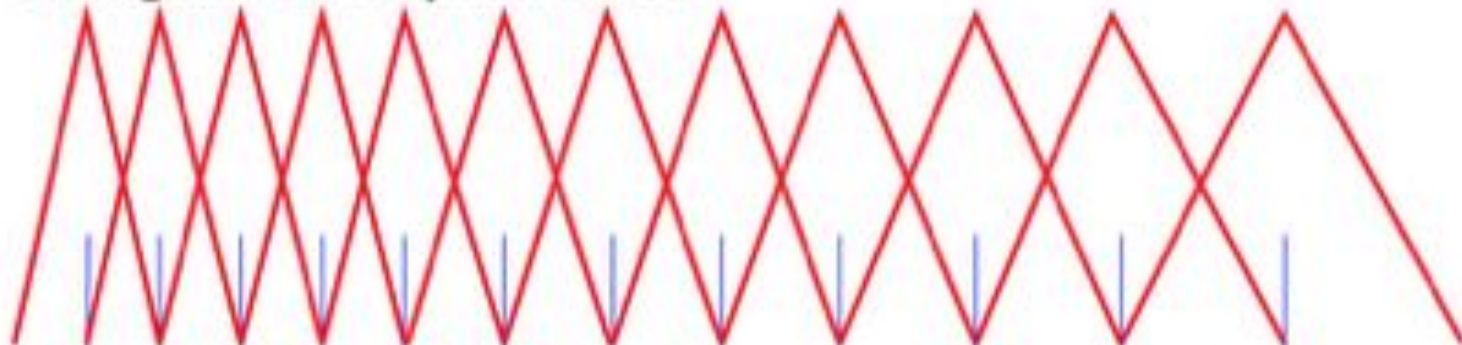


DFT(STFT) power spectrum $|X[k]|^2$



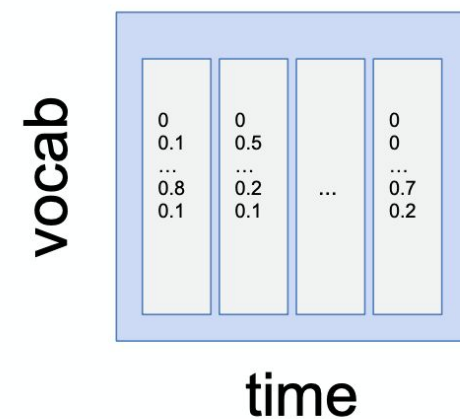
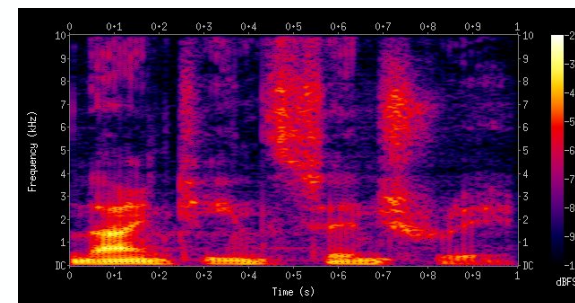
→ Frequency bins

Triangular band-pass filters

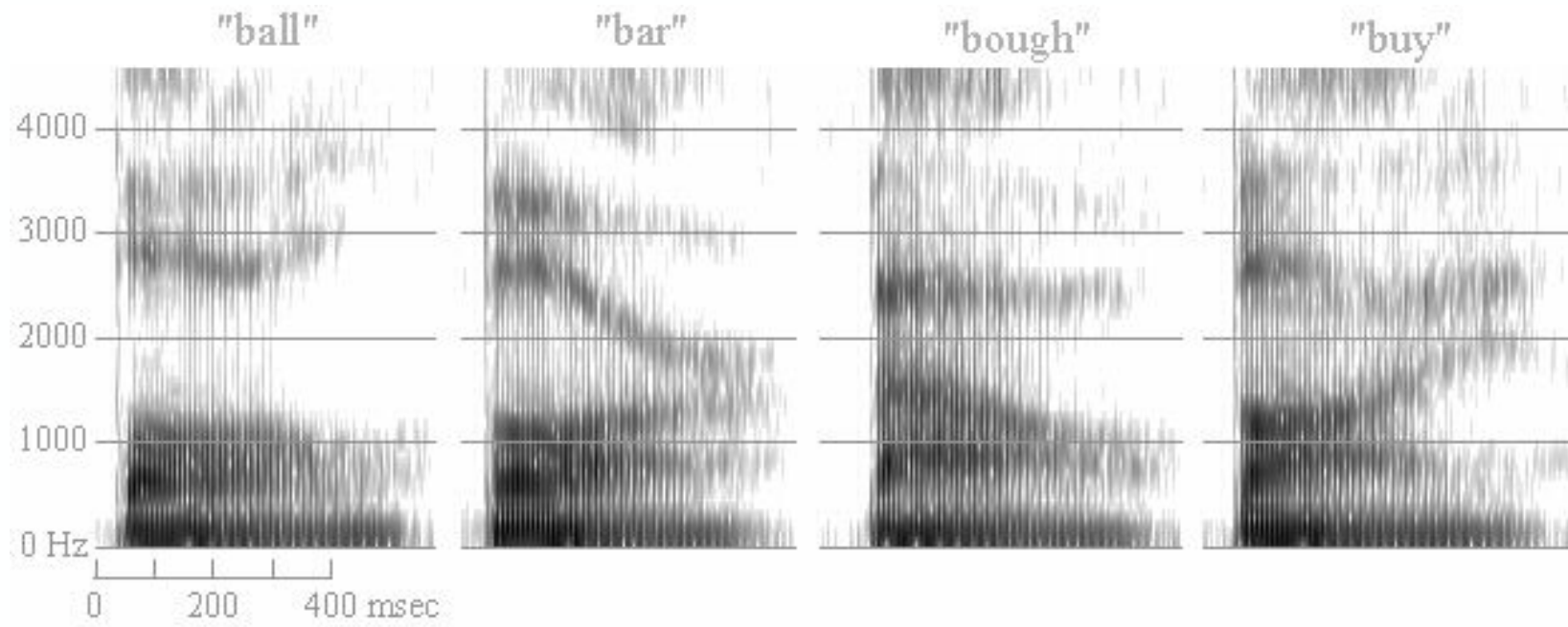


Mel-scale power spectrum $Y[m]$

- Препроцессинг
- **Акустическая модель**
- Языковая модель
- Пунктуационная модель



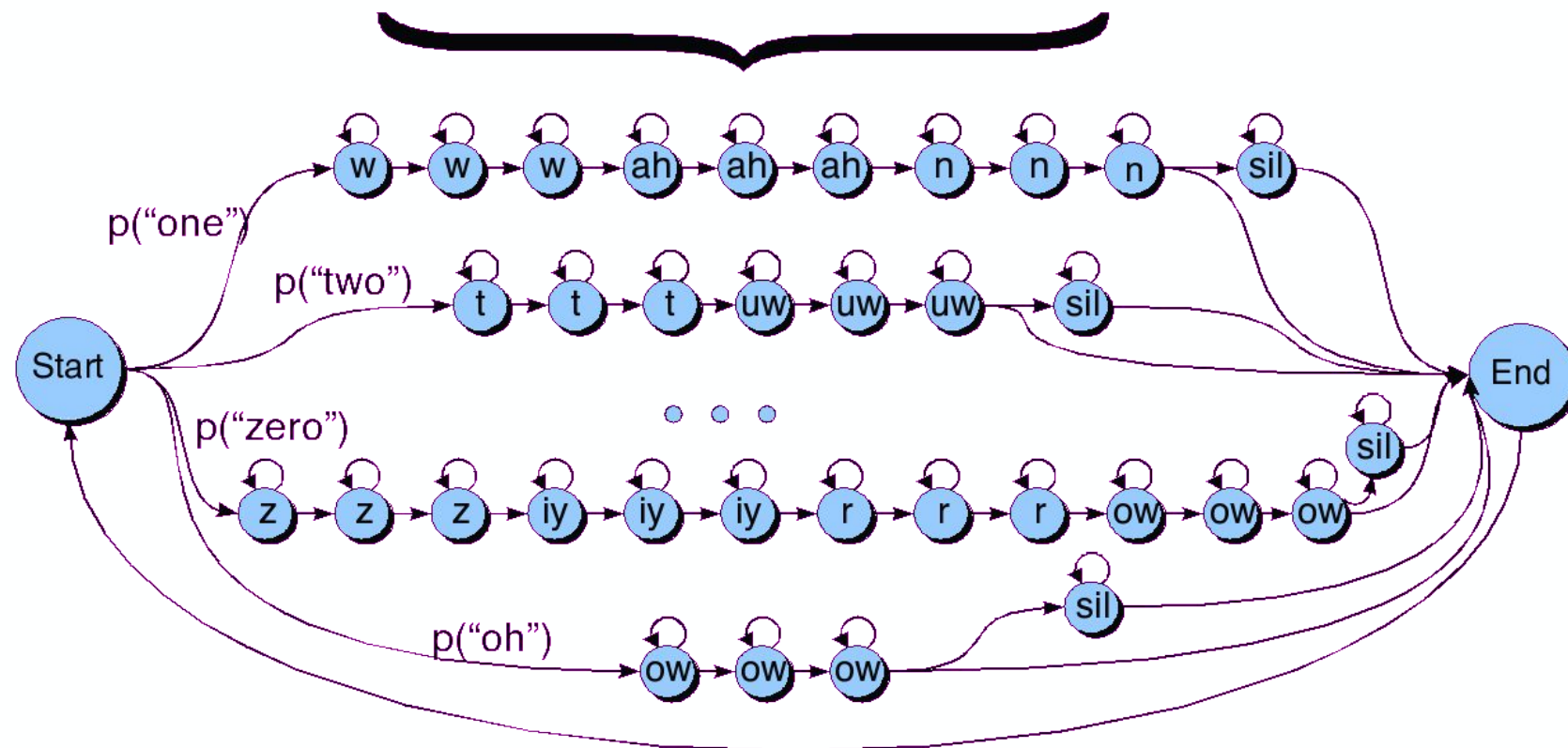
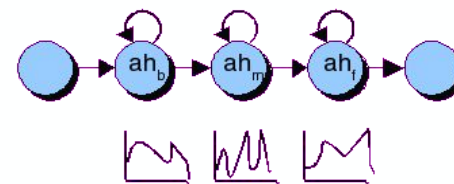
1950-1960 сейсмограммы



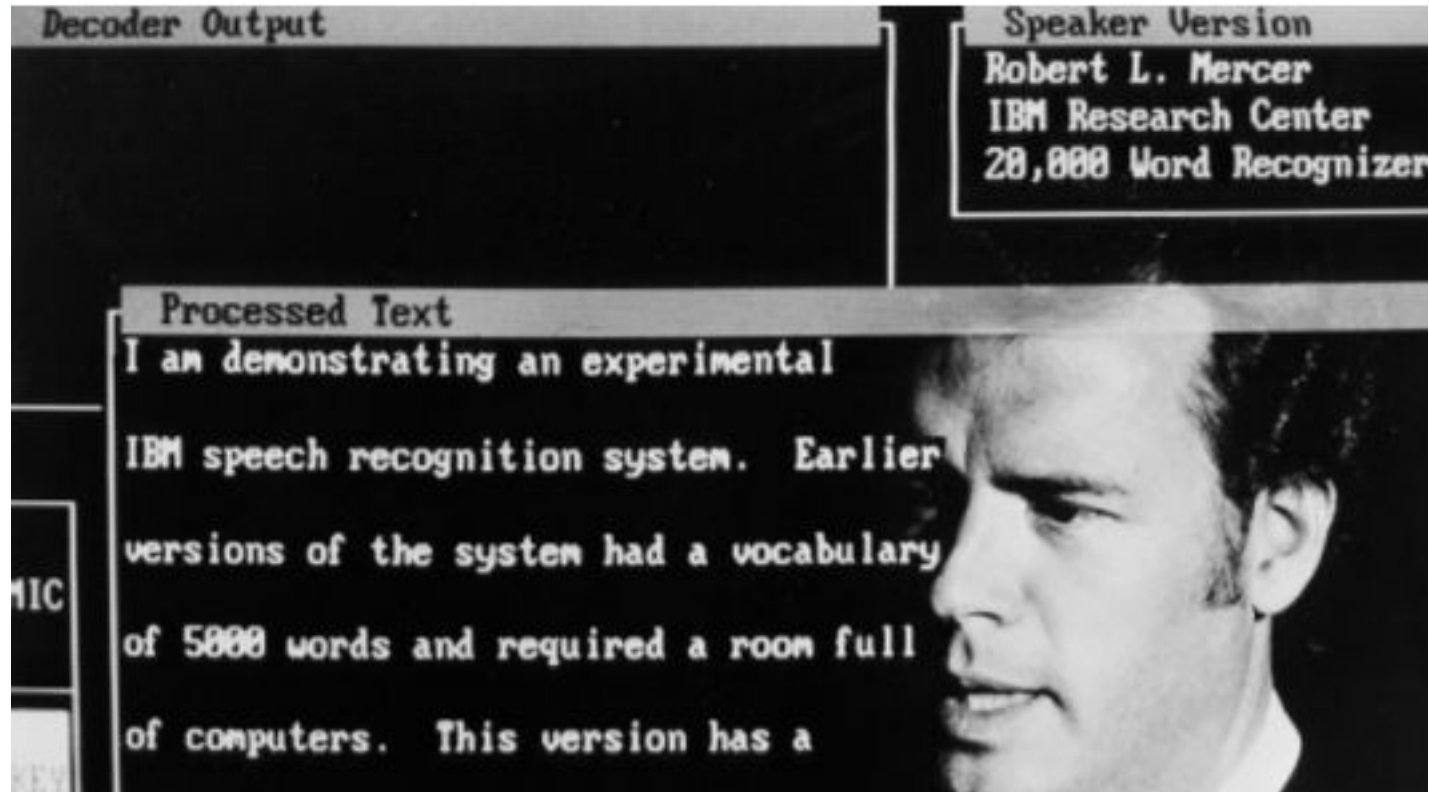
Lexicon

one	w ah n
two	t uw
three	th r iy
four	f ao r
five	f ay v
six	s ih k s
seven	s eh v ax n
eight	ey t
nine	n ay n
zero	z iy r ow
oh	ow

Phone HMM



1970-1990



CMU Sphinx
Project by Carnegie Mellon University

[DOWNLOAD](#) [TUTORIAL](#) [WIKI](#) [DEVELOP](#) [RESEARCH](#) [ABOUT](#)



Pocketsphinx

С 2009 года в задаче распознавания речи применяются нейронные сети

Метрики качества

word accuracy

какого дьявола ты здесь шумишь

какого дявола ты здесь шумишь

word accuracy

какого дьявола ты здесь шумишь

какого дьявола ты здесь шумишь

problems?

WER, CER, SER

какого дьявола ты здесь шумишь

какого дьявола ты здесь шумишь

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

where...

S = number of substitutions

D = number of deletions

I = number of insertions

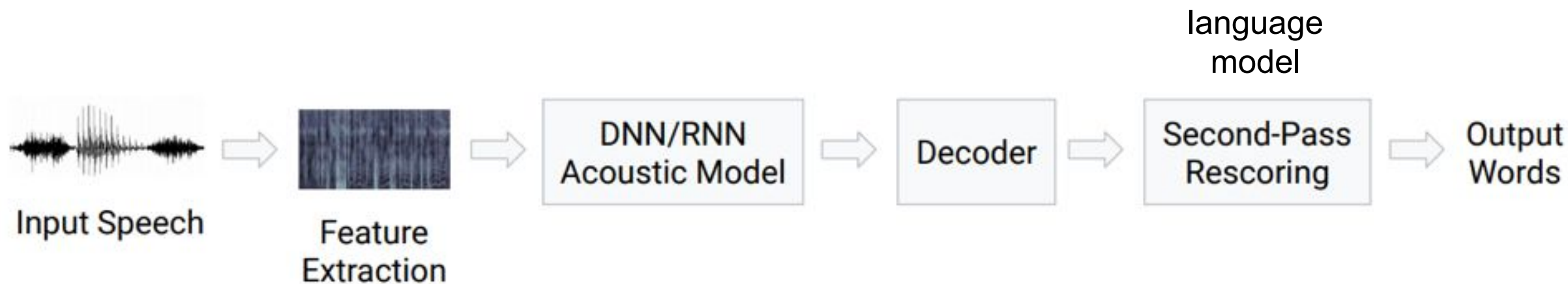
N = number of words in the reference
chars
sentences

ASR pipeline

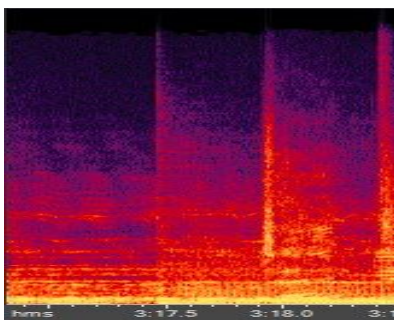
(Hybrid ASR)

Conventional ASR

Pipeline



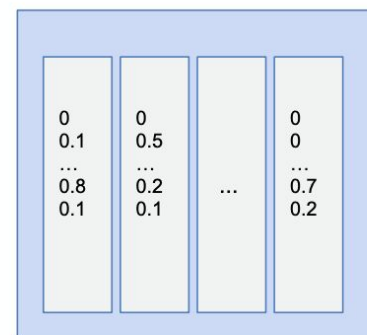
Акустическая модель



Acoustic model

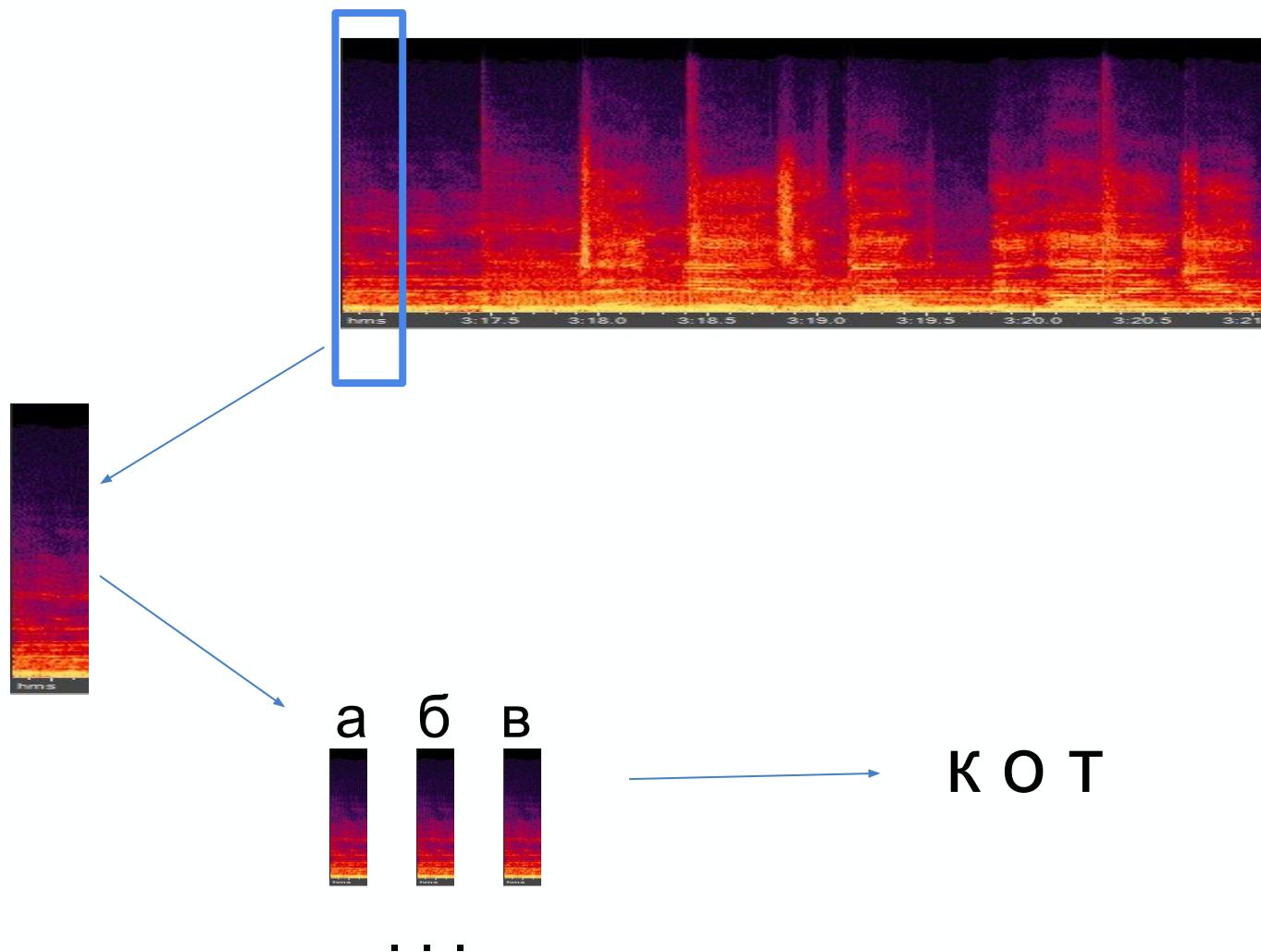


vocab



time

frame-level prediction?



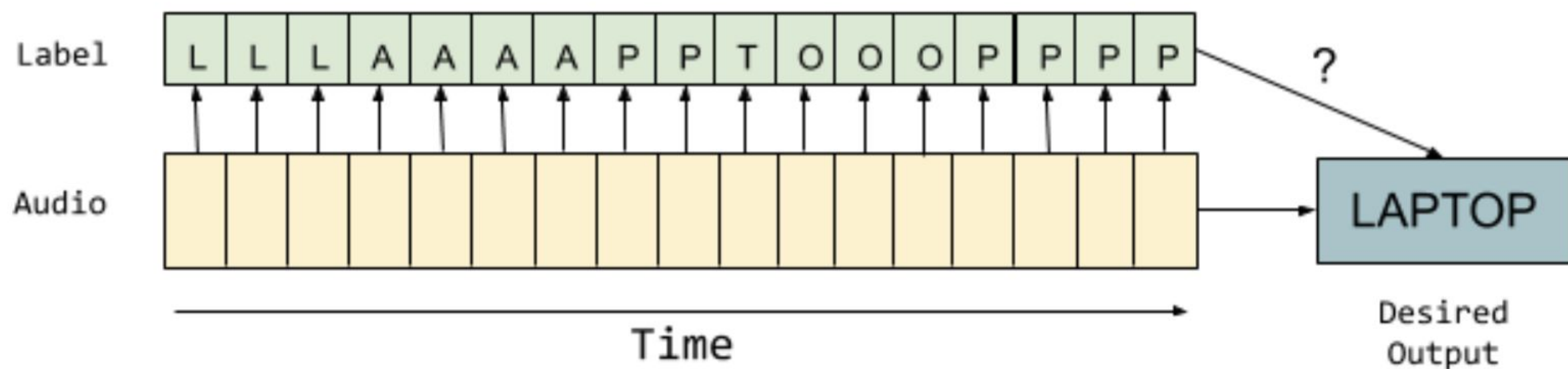


Figure 1. Here, we have aligned audio data, where the audio is chopped up into time slices and each is labeled with a letter. But it's very difficult to go from those labels to the correct transcript, especially considering words with repeated letters (such as "book").

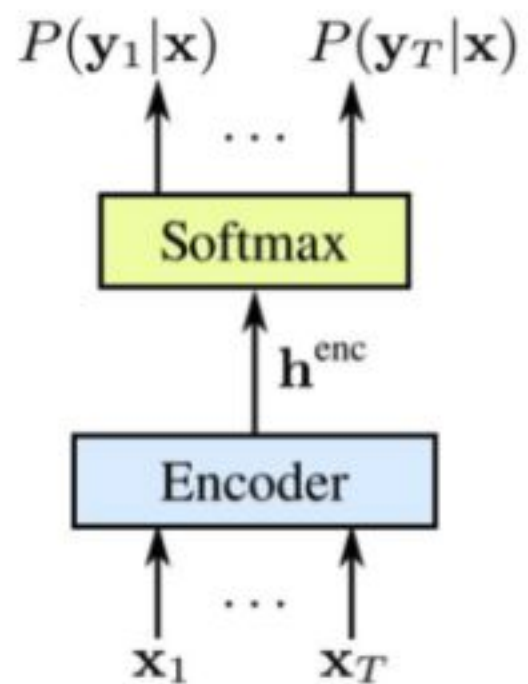
$\log (\text{Pr} (\text{output: "BOOK"} \mid \text{audio})) = \log (\text{Pr} (\text{BOO-OOO} - \text{KK} \mid \text{audio})) + \log (\text{Pr} (\text{BBO} - \text{OO-KKK} \mid \text{audio})) + \dots).$

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

__кккллкласс__сс__



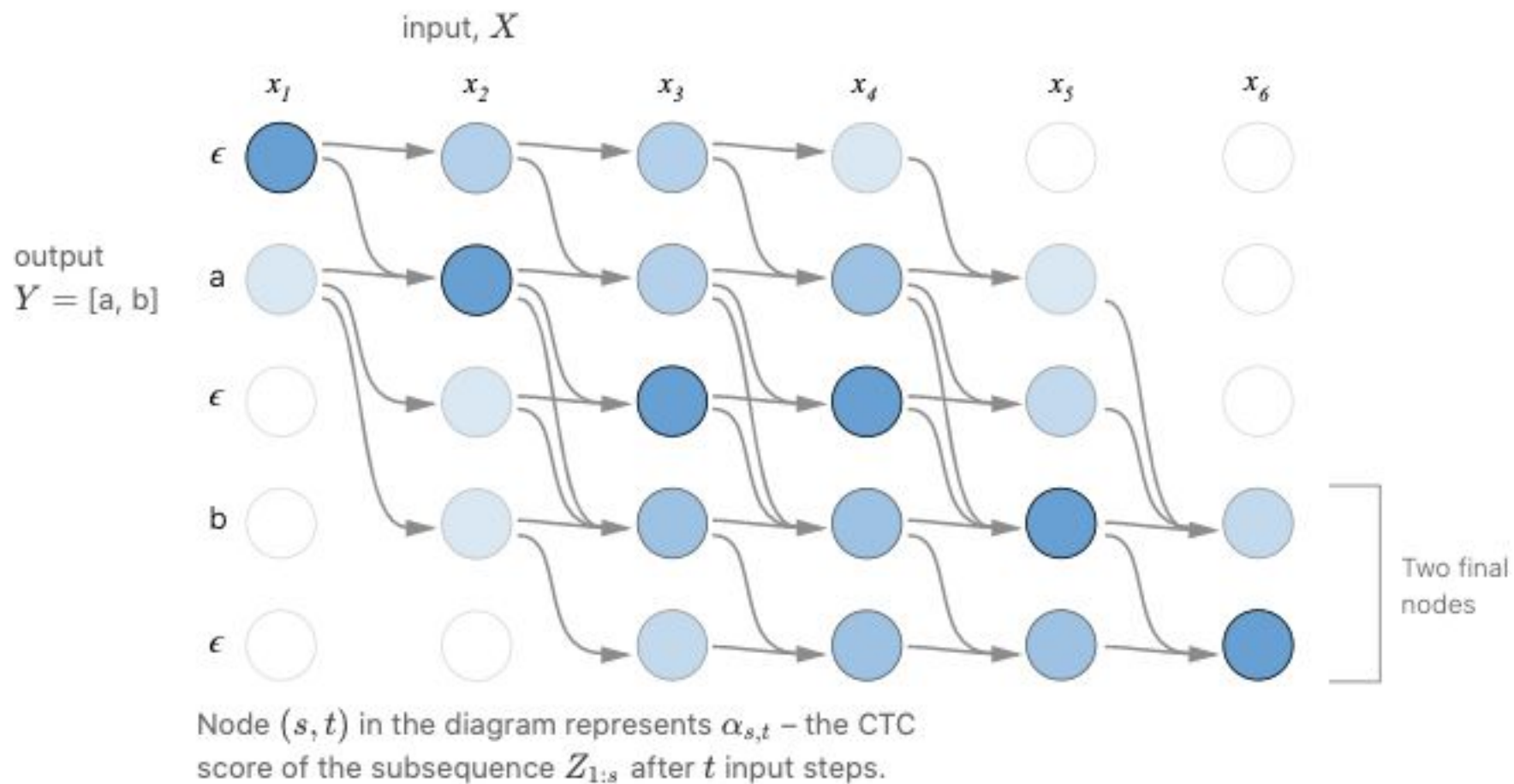
класс



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



t h e q u i c k b r o w n f o x



The quick brown fox

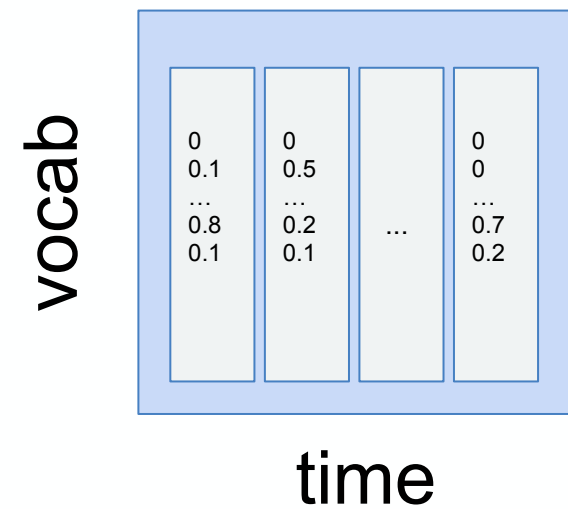
Handwriting recognition: The input can be (x, y) coordinates of a pen stroke or pixels in an image.

j u m p s o v e r t h e l a z y d o g

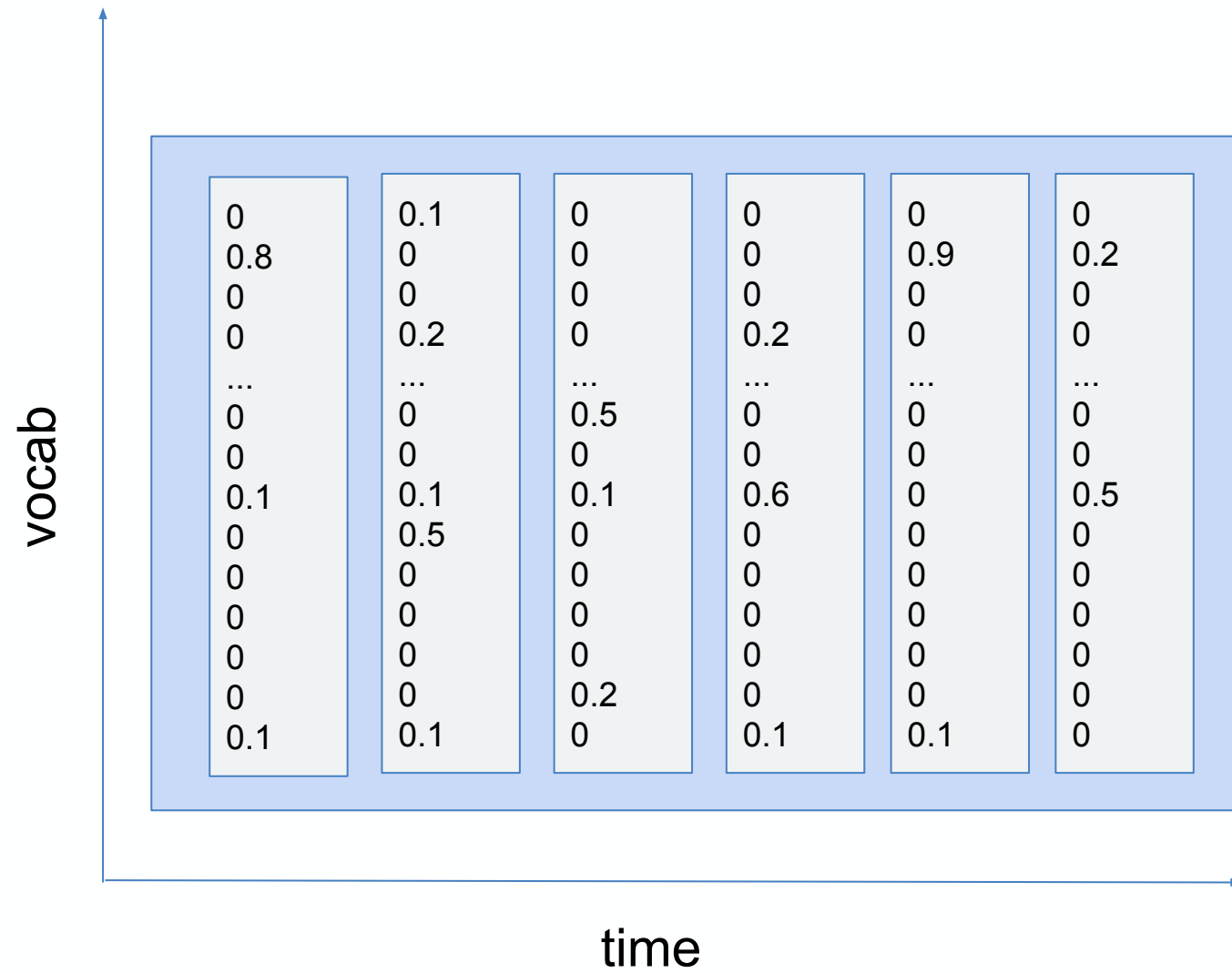


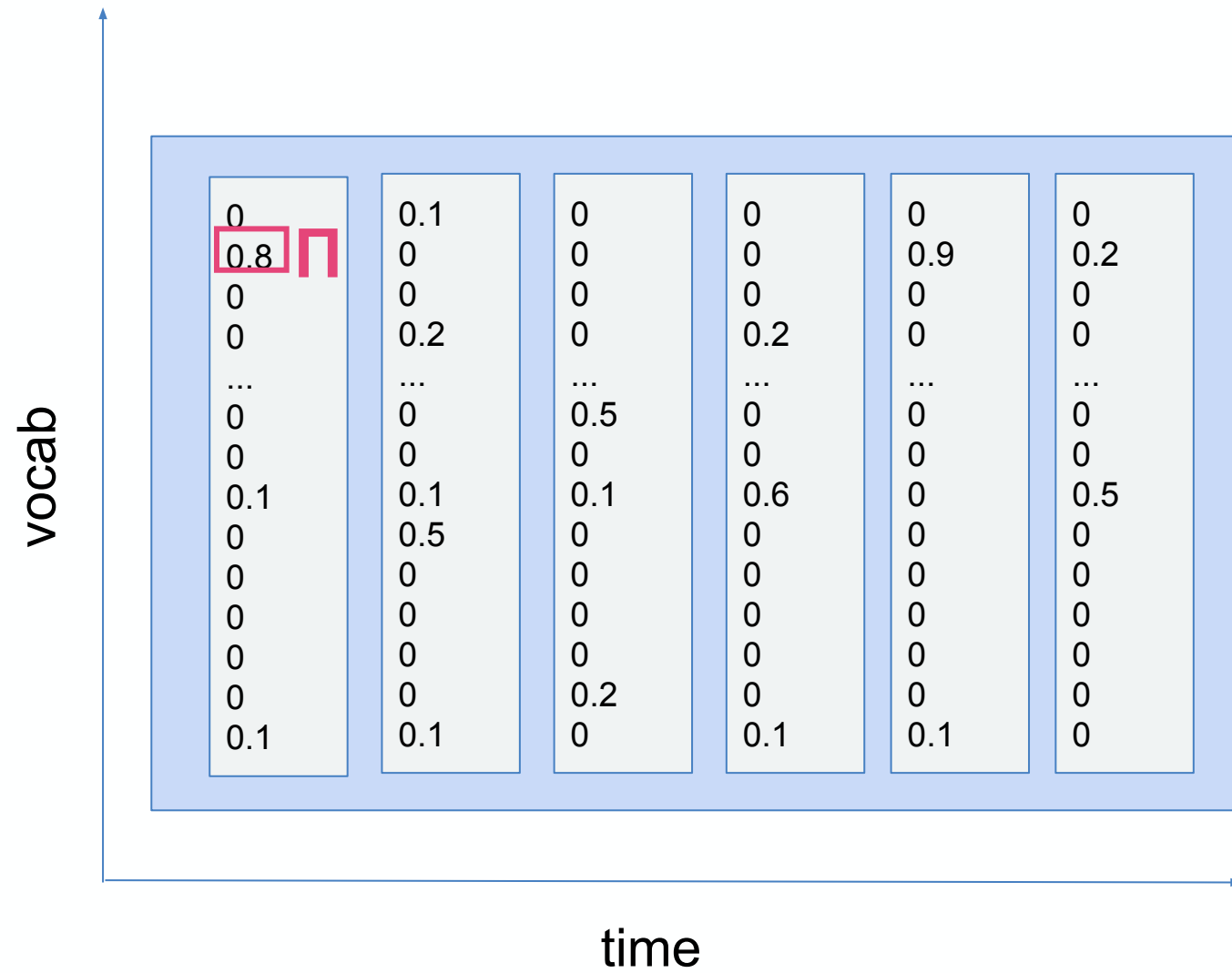
Speech recognition: The input can be a spectrogram or some other frequency based feature extractor.

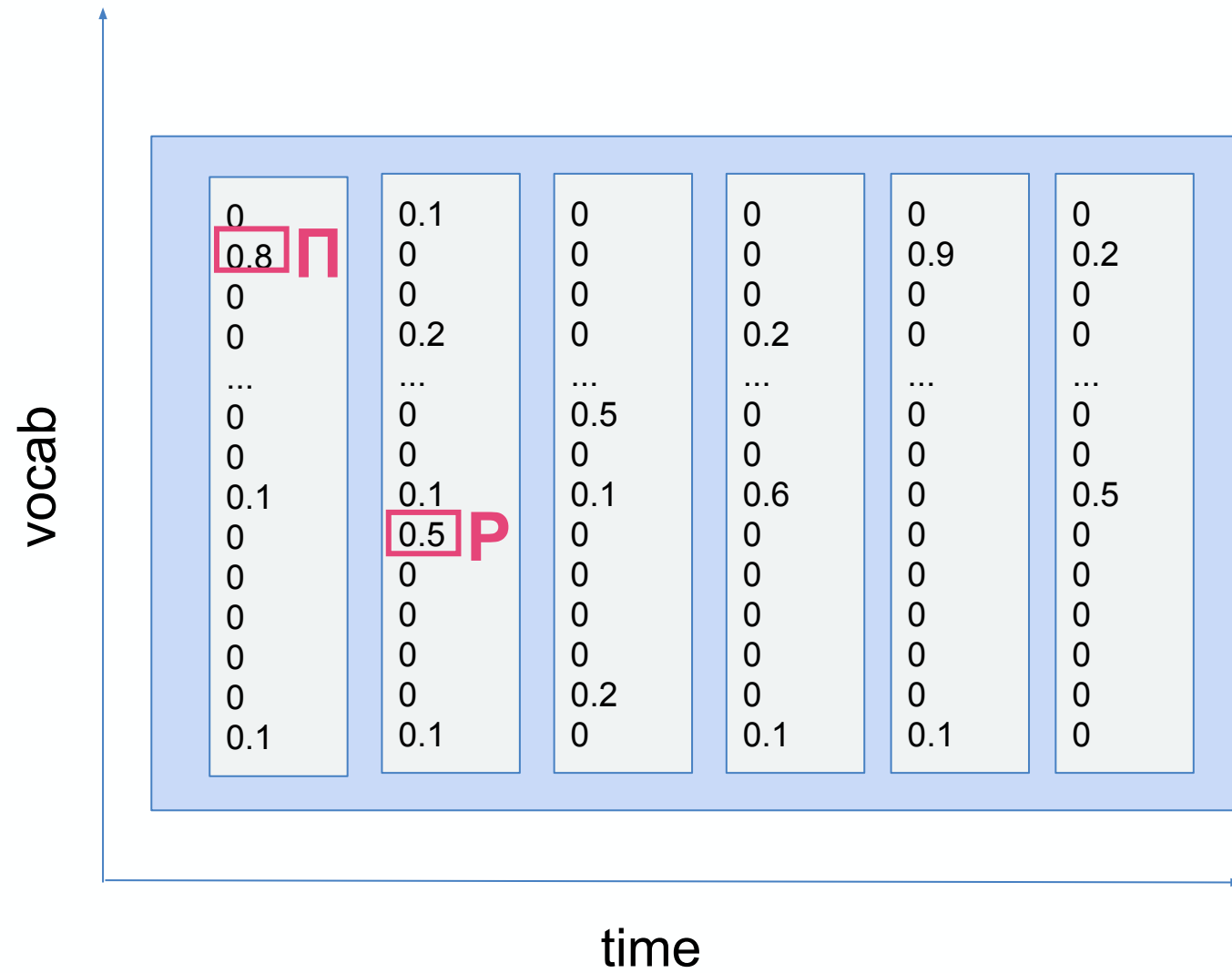
- Препроцессинг
- Акустическая модель
- **Языковая модель**
- Пунктуационная модель

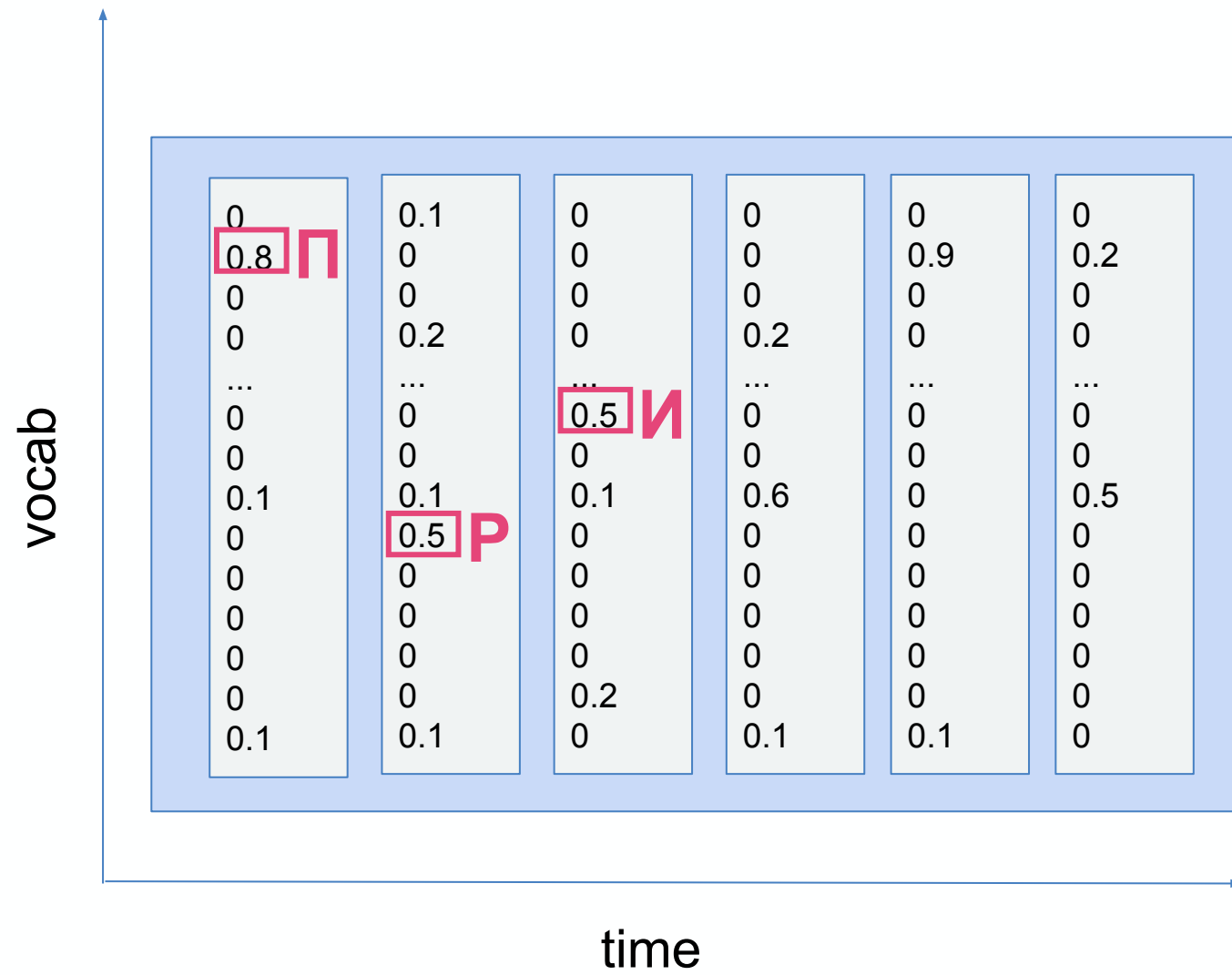


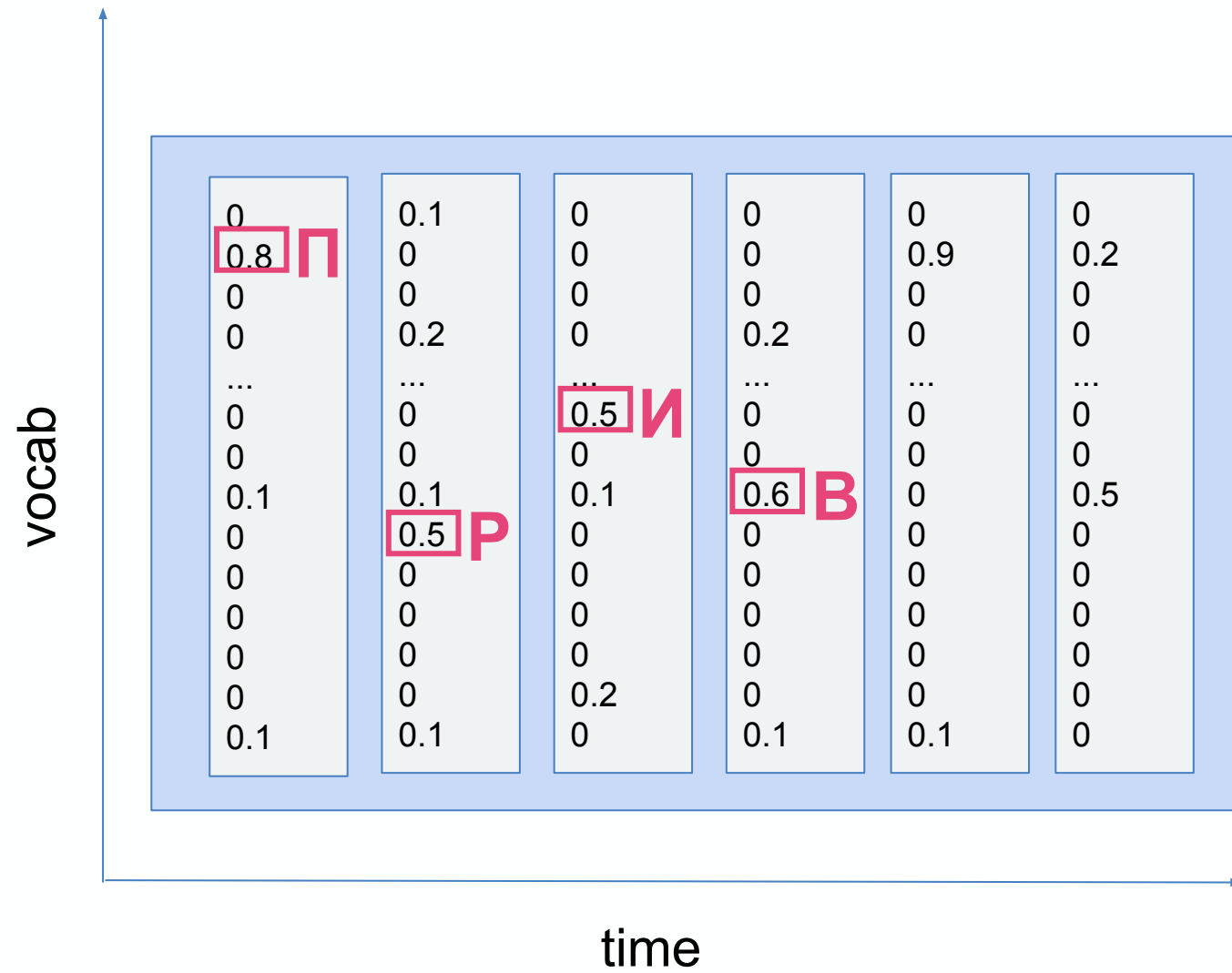
**привет соня пойдем
сегодня в кинчик**

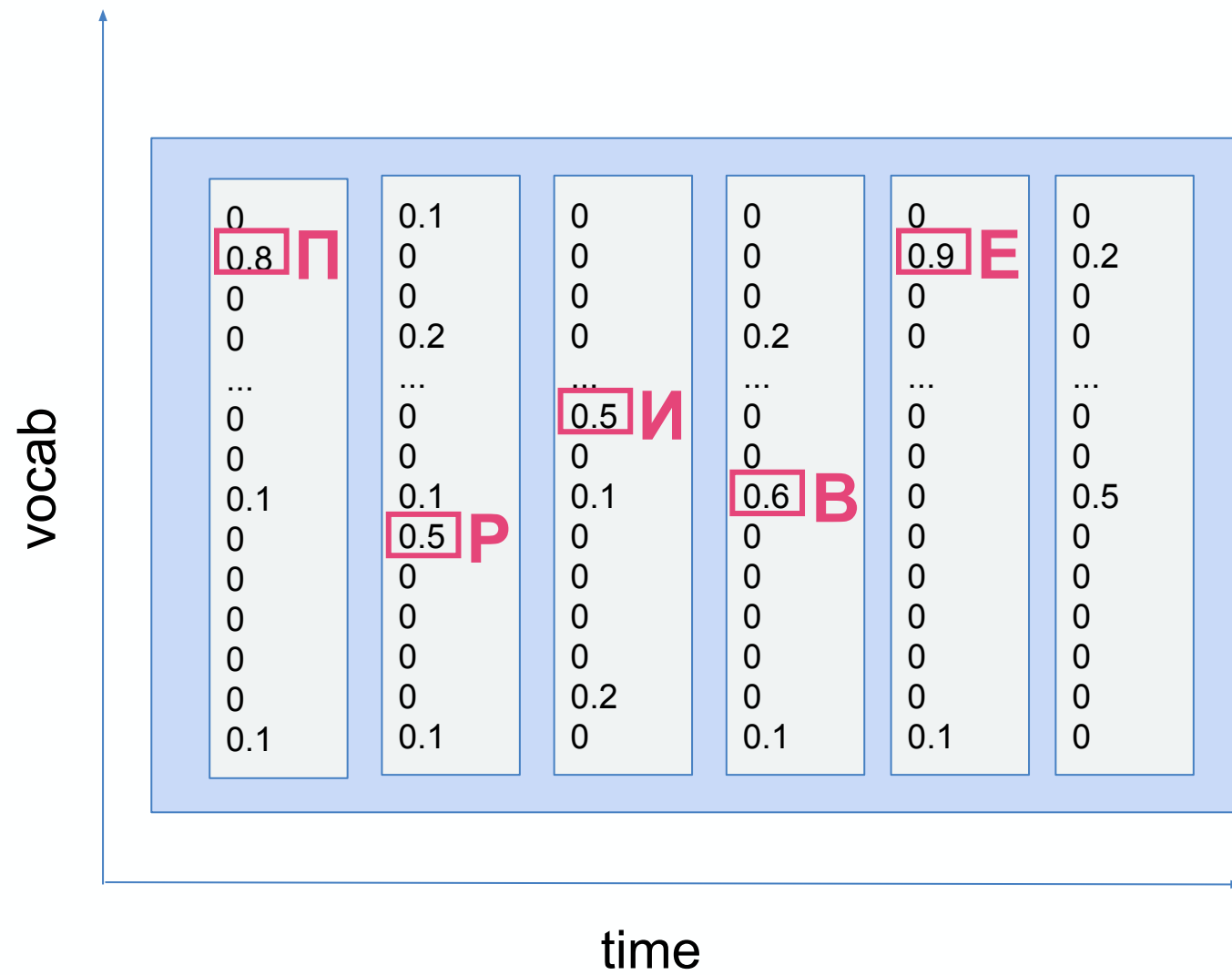


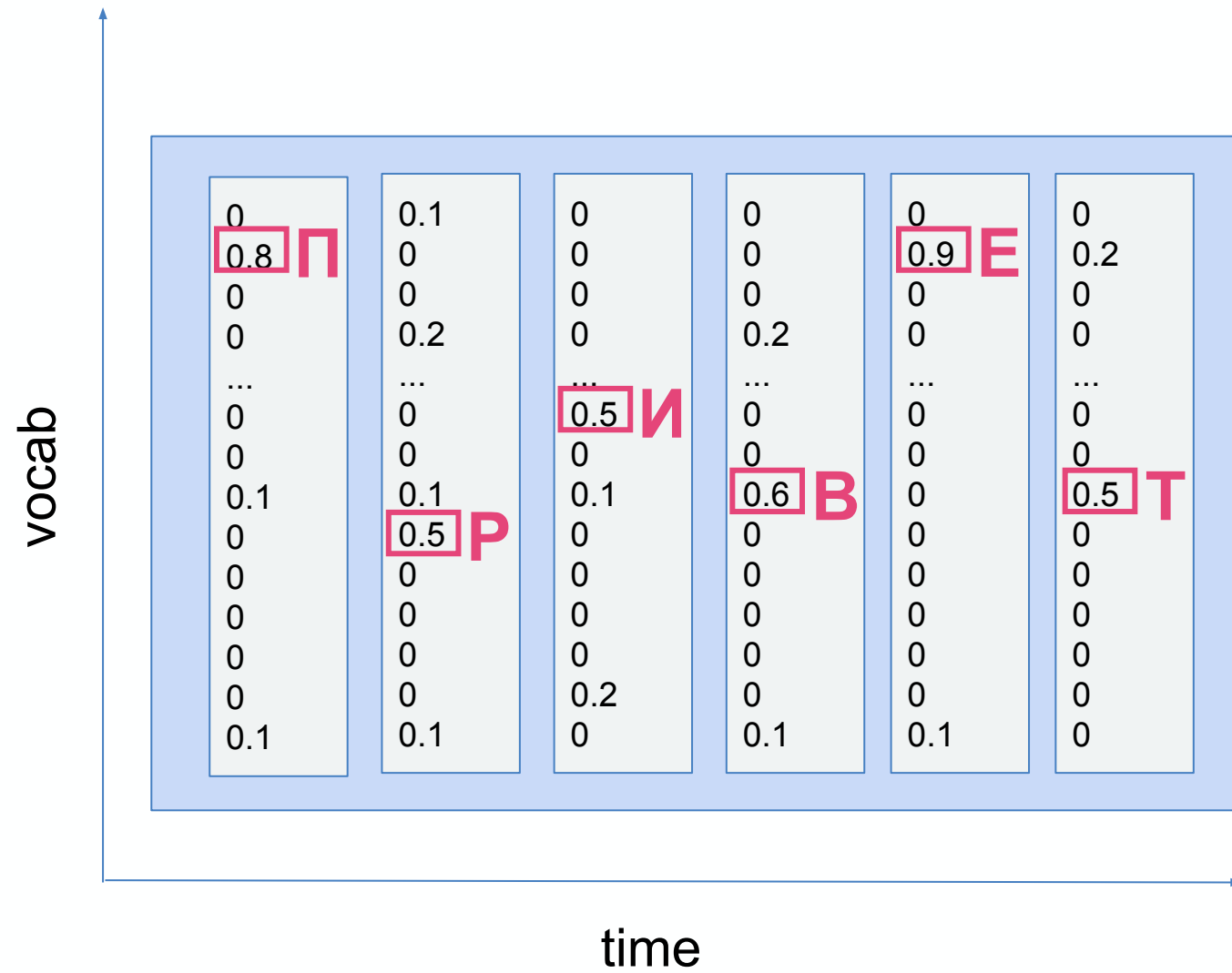


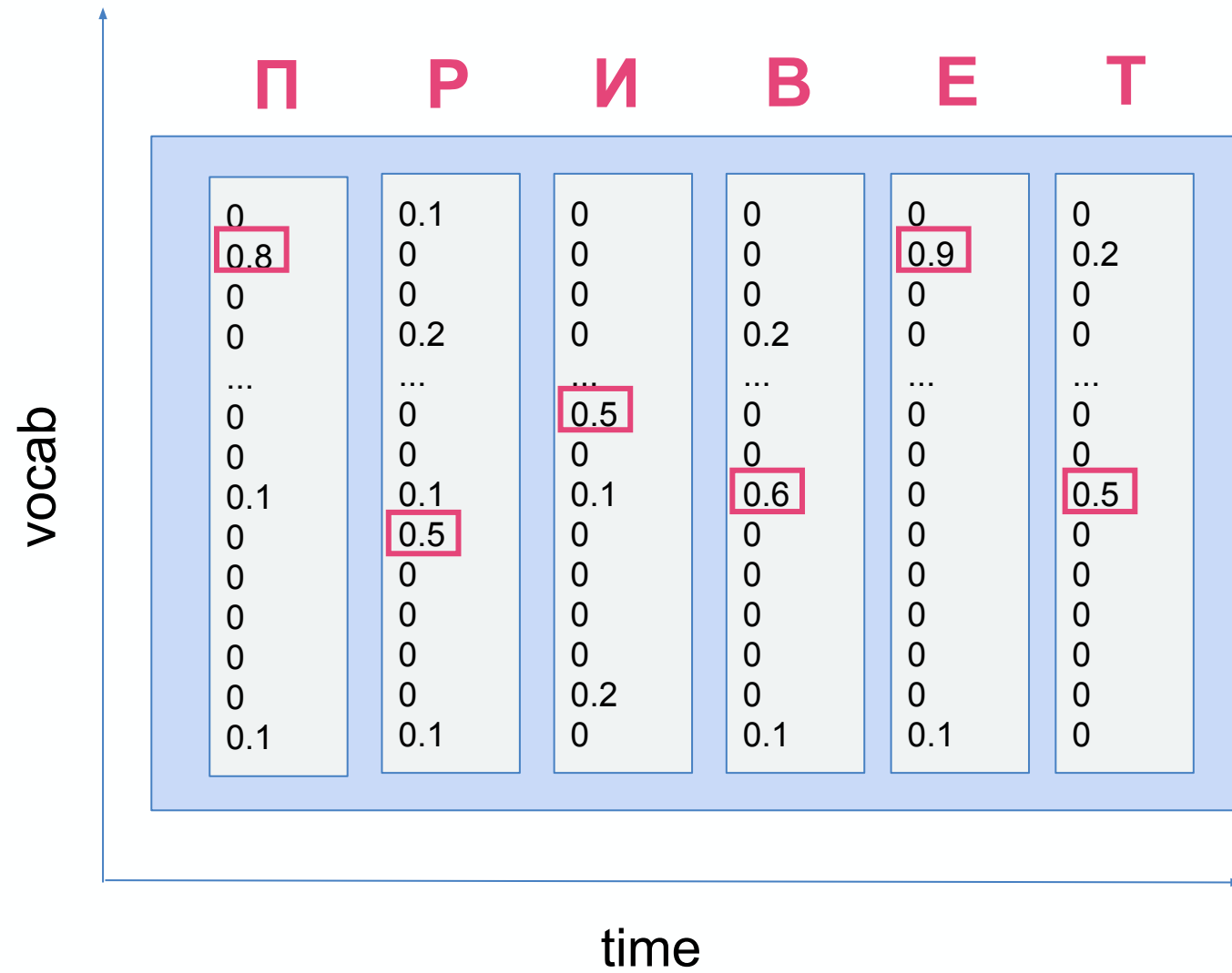




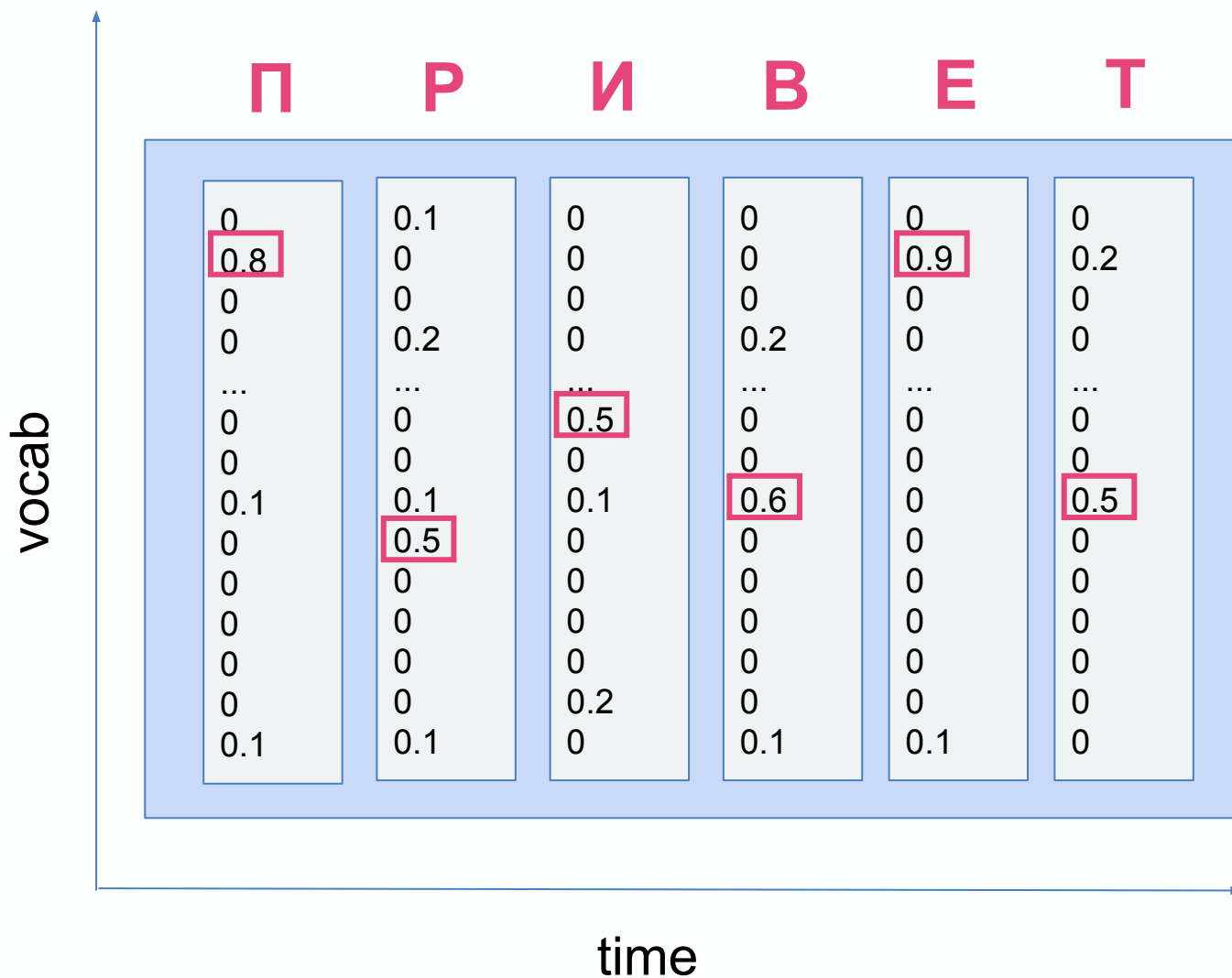








greedy decoder



greedy decoder

ПРОБЛЕМЫ?

greedy decoder

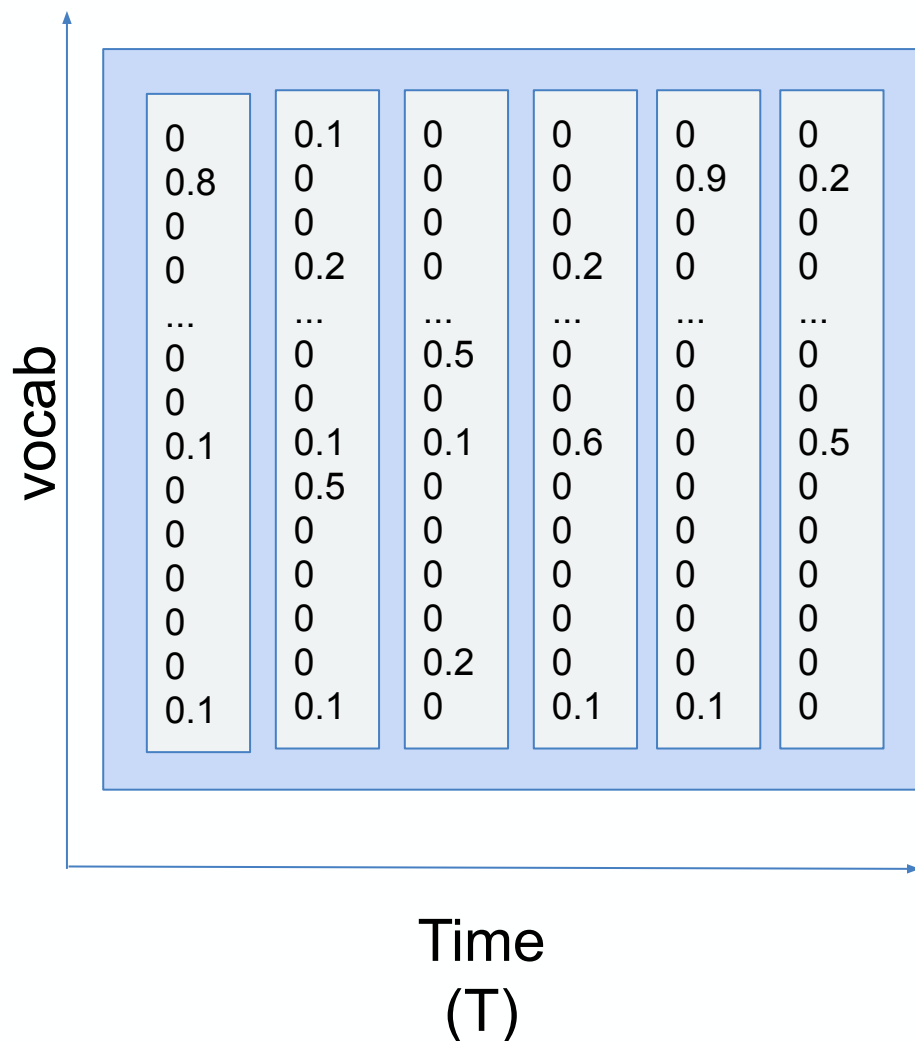
ПРОБЛЕМЫ?

- CTC loss возвращает frame-by-frame prediction
- Несколько интерпретаций:
“привет”: ‘-пприии--
вет’, ‘прииве-т’, ‘пр--
ивет’...

basic beam search decoder

Beam search decoding итеративно
создает кандидатов (beams)
и присваивает им скоры

basic beam search decoder

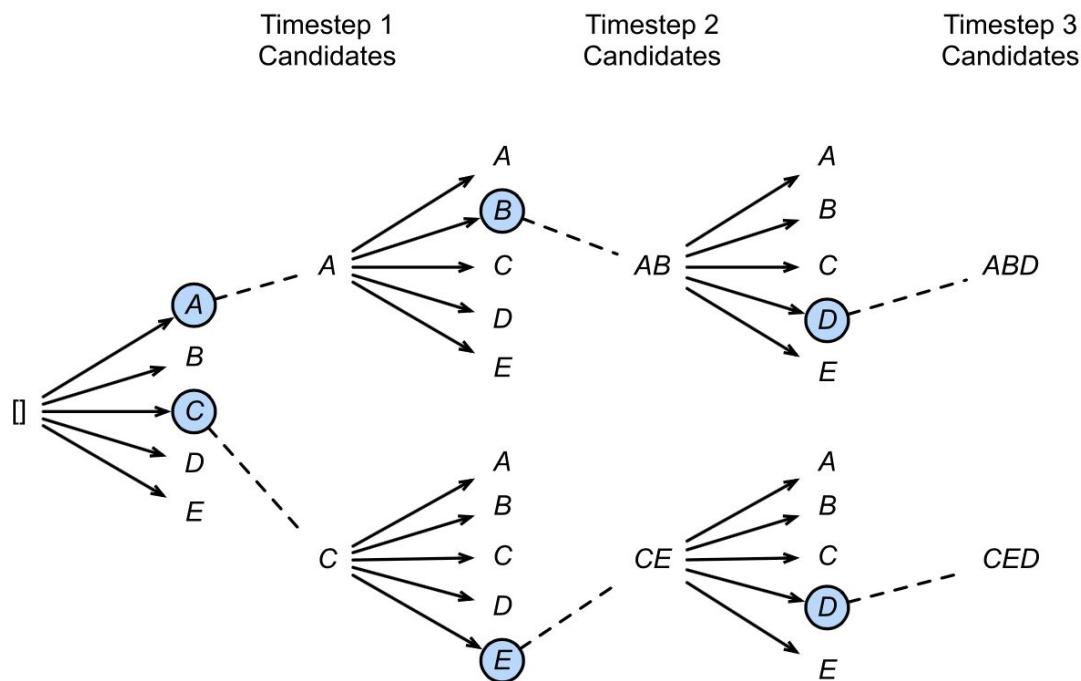


Data: NN output matrix mat , BW

Result: decoded text

```
1  $beams = \{\emptyset\}$ ;
2  $scores(\emptyset, 0) = 1$ ;
3 for  $t = 1 \dots T$  do
4    $bestBeams = bestBeams(beams, BW)$ ;
5    $beams = \{\}$ ;
6   for  $b \in bestBeams$  do
7      $beams = beams \cup b$ ;
8      $scores(b, t) = calcScore(mat, b, t)$ ;
9     for  $c \in alphabet$  do
10       $b' = b + c$ ;
11       $scores(b', t) = calcScore(mat, b', t)$ ;
12       $beams = beams \cup b'$ ;
13    end
14  end
15 end
16 return  $bestBeams(beams, 1)$ ;
```

basic beam search decoder



Data: NN output matrix mat , BW

Result: decoded text

```

1  $beams = \{\emptyset\}$ ;
2  $scores(\emptyset, 0) = 1$ ;
3 for  $t = 1 \dots T$  do
4    $bestBeams = bestBeams(beams, BW)$ ;
5    $beams = \{\}$ ;
6   for  $b \in bestBeams$  do
7      $beams = beams \cup b$ ;
8      $scores(b, t) = calcScore(mat, b, t)$ ;
9     for  $c \in alphabet$  do
10       $b' = b + c$ ;
11       $scores(b', t) = calcScore(mat, b', t)$ ;
12       $beams = beams \cup b'$ ;
13    end
14  end
15 end
16 return  $bestBeams(beams, 1)$ ;

```

basic beam search decoder

$$\log P_{AM}(\hat{\mathbf{y}}|\mathbf{x}) + \alpha \log P_{LM}(\hat{\mathbf{y}})$$

Data: NN output matrix mat , BW

Result: decoded text

```
1 beams = {∅};
2 scores(∅, 0) = 1;
3 for t = 1...T do
4     bestBeams = bestBeams(beams, BW);
5     beams = {};
6     for b ∈ bestBeams do
7         beams = beams ∪ b;
8         scores(b, t) = calcScore(mat, b, t);
9         for c ∈ alphabet do
10             b' = b + c;
11             scores(b', t) = calcScore(mat, b', t);
12             beams = beams ∪ b';
13         end
14     end
15 end
16 return bestBeams(beams, 1);
```

basic beam search decoder

$$\log P_{AM}(\hat{\mathbf{y}}|\mathbf{x}) + \alpha \log P_{LM}(\hat{\mathbf{y}})$$

Data: NN output matrix mat , BW

Result: decoded text

```
1  $beams = \{\emptyset\}$ ;  
2  $scores(\emptyset, 0) = 1$ ;  
3 for  $t = 1 \dots T$  do  
4    $bestBeams = bestBeams(beams, BW)$ ;  
    $beams = \{\}$ ;  
   for  $b \in bestBeams$  do  
      $beams = beams \cup b$ ;  
      $scores(b, t) = calcScore(mat, b, t)$ ;  
     for  $c \in alphabet$  do  
        $b' = b + c$ ;  
        $scores(b', t) = calcScore(mat, b', t)$ ;  
        $beams = beams \cup b'$ ;  
     end  
   end  
15 end  
16 return  $bestBeams(beams, 1)$ ;
```

ground truth

привет сонь пойдём сегодня в кинчик

greedy

при вет сня пойдёмсегодня в кичик

beam search LM decoder

привет соня пойдём сегодня в кинчик

Ground truth: "the fake friend of the family, like the"
Best path decoding: "the fak friend of the fomly hae tC"
Beam search: "the fak friend of the fomcly hae tC"
Beam search with LM: "the fake friend of the family, lie th"

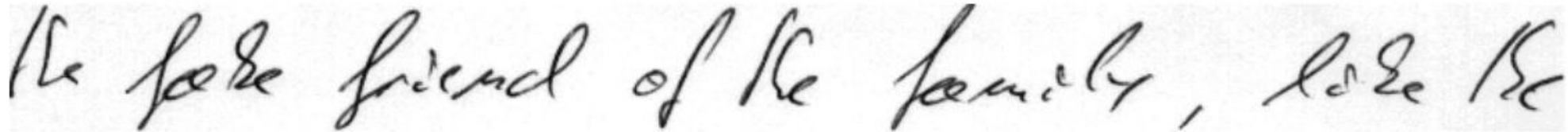
A sample of handwritten text from the IAM dataset. The text is written in a cursive script and reads: "the fake friend of the family, like the". The handwriting is somewhat informal and slightly slanted.

Fig. 8: Sample from IAM dataset.

Neural Language Models

Model	dev		test	
	clean	other	clean	other
baseline (100-best)	7.17	19.79	7.26	20.37
GPT-2 (117M, cased)	5.39	16.81	5.64	17.60
BERT (base, cased)	5.17	16.44	5.41	17.41
RoBERTa (base, cased)	5.03	16.16	5.25	17.18
GPT-2 (345M, cased)	5.15	16.48	5.30	17.26
BERT (large, cased)	4.96	16.26	5.25	16.97
RoBERTa (large, cased)	4.75	15.81	5.05	16.79
oracle (100-best)	2.85	12.21	2.81	12.85

Table 1: WERs on LibriSpeech after reranking. Baseline lists and oracle numbers are from [Shin et al. \(2019\)](#).

n-gram model

Bigrams:

“какого дьявола ты здесь шумишь”

$$P = P(\text{какого}) * P(\text{дьявола}|\text{какого}) * P(\text{ты}|\text{дьявола}) * \\ P(\text{здесь}|\text{ты}) * P(\text{шумишь}|\text{здесь})$$

1. $P(\text{дьявола}) = \# \text{”дьявола”} / \# \text{слов}$
2. $P(\text{дьявола}|\text{какого}) = \# \text{”дьявола-какого”} / \# \text{”какого”}$

n-gram model

ПРОБЛЕМЫ?

n-gram model

ПРОБЛЕМЫ?

- Десятки of GB!
- Denormalized corpus (24 сентября 2020, и тд, в 90 годы, в 3 из 4 случаев, etc)

n-gram model

ПРОБЛЕМЫ?

- binarize!
- normalize!

text normalization

в первый раз в девяностых она купила один литр
молока и так далее за двадцать рублей

text normalization

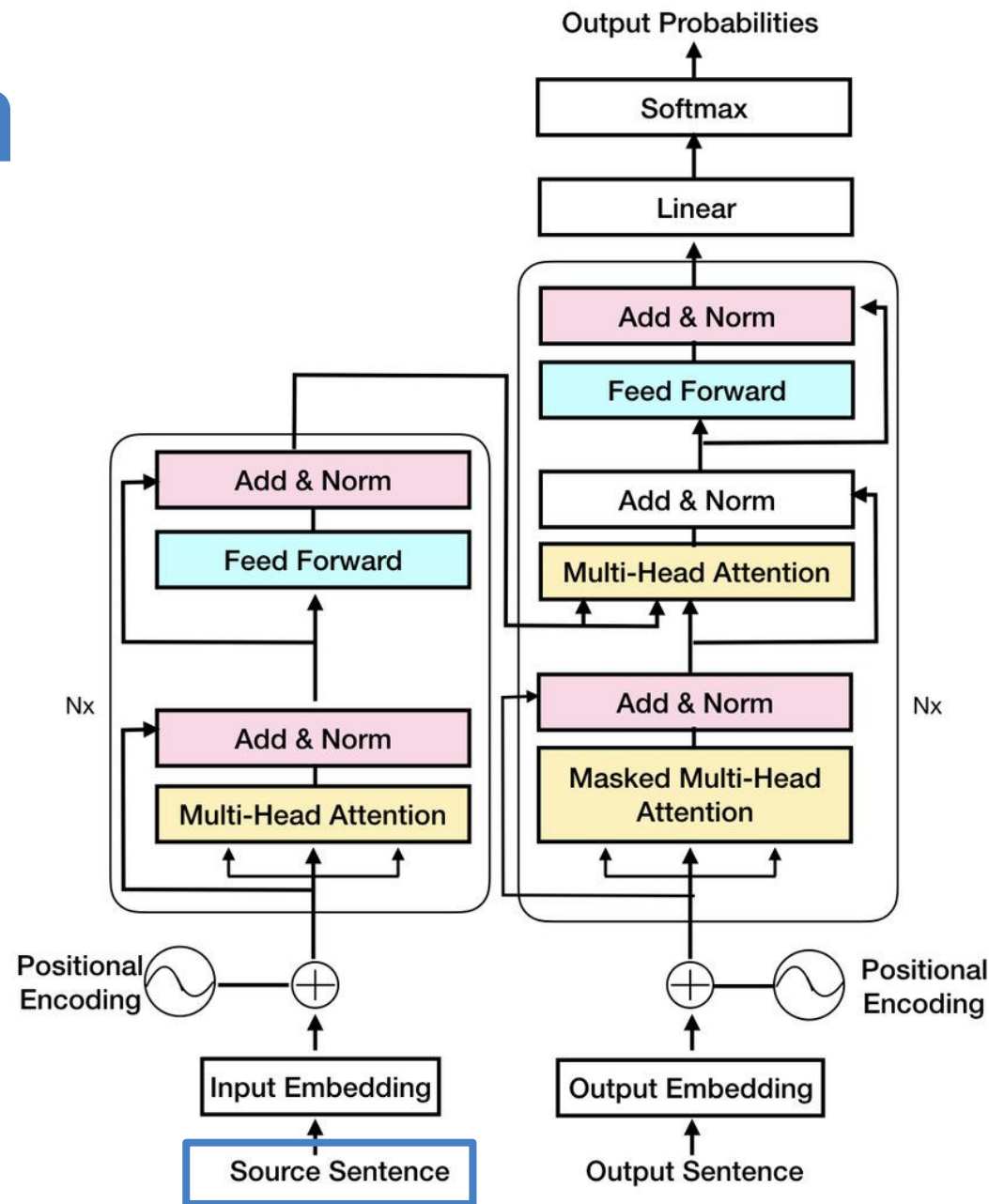
в 1 раз в 90 она купила 1 л молока и т д за 20 руб

text normalization

в 1 раз в 90 она купила 1 л
молока и т д за 20 руб

youtokentome

pre-trained embeddings



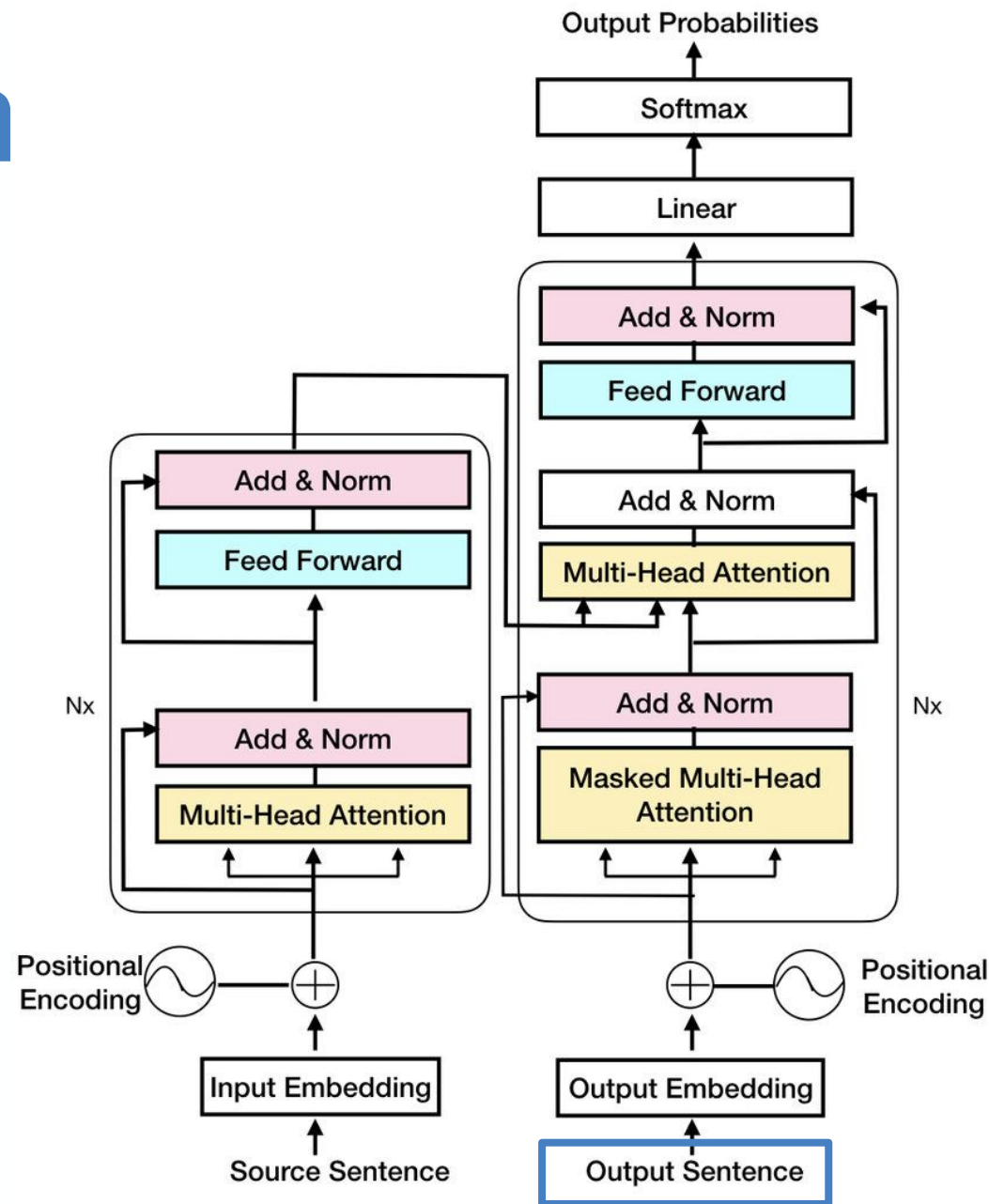
text normalization

в первый раз в девяностых
она купила один литр молока
и так далее за двадцать
рублей

youtokentome

pre-trained embeddings

https://www.researchgate.net/publication/338223294_Neural_Machine_Translation_for_the_Bangla-English_Language_Pair/figures?lo=1



text normalization



Надя 09:19

эх лихие 90



act_normal 09:19

эх лихие девяностые



Надя 09:21

У нас было 2 пакетика травы 75 ампул мескалина 5 пакетиков диэтиламида лизергиновой кислоты или лсд солонка наполовину наполненная кокаином и целое море разноцветных амфетаминов барбитуратов и транквилизаторов а так же литр текилы литр рома ящик бадвайзера пинта чистого эфира и 12 пузырьков амилнитрита не то чтобы все это было категорически необходимо в поездке но если уж начал собирать коллекцию то к делу надо подходить серьезно



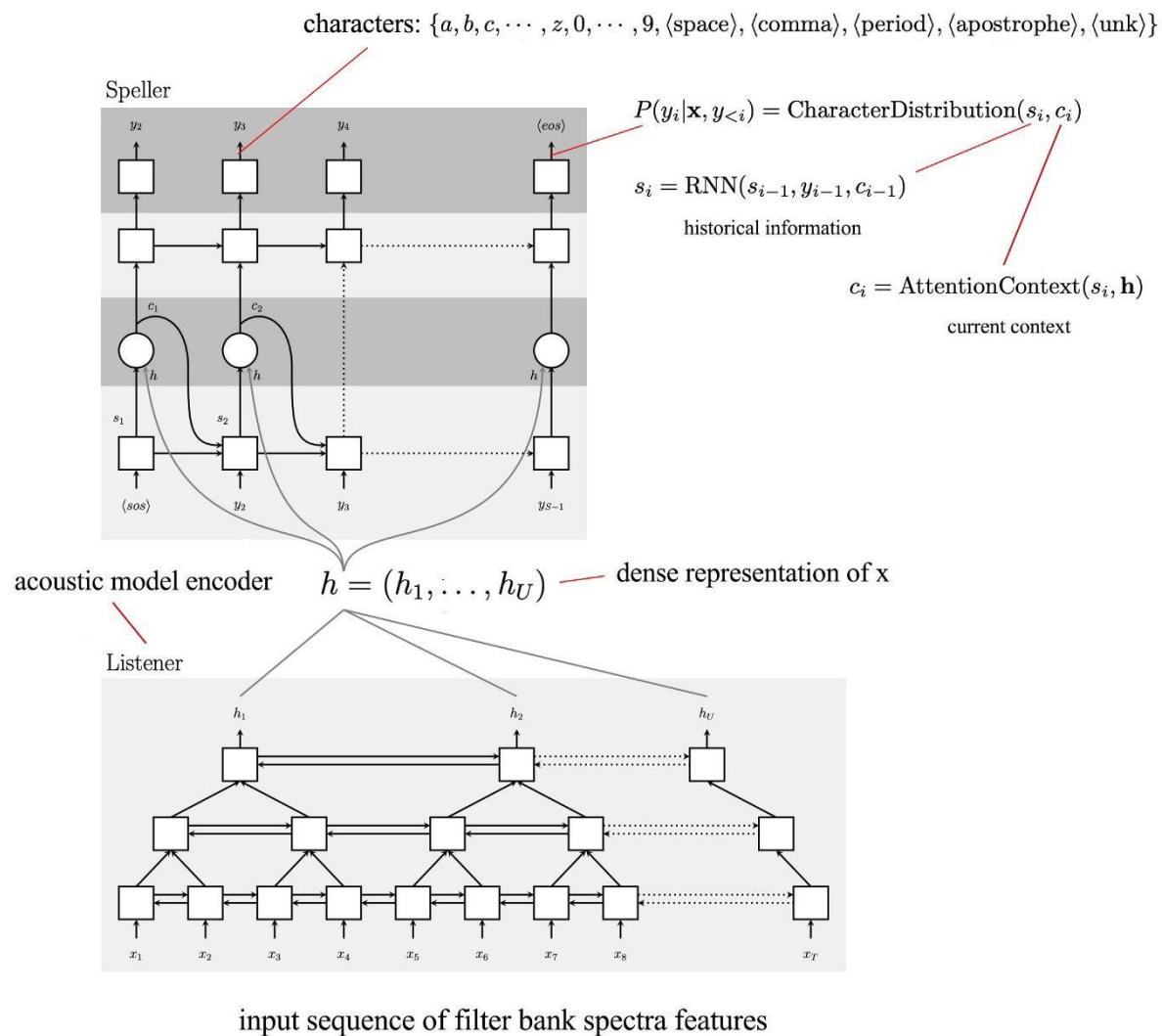
act_normal 09:21

у нас было два пакетика травы семьдесят пять ампул мескалина пять пакетиков диэтиламида лизергиновой кислоты или лсд солонка наполовину наполненная кокаином и целое море разноцветных амфетаминов барбитуратов и транквилизаторов а так же литр текилы литр рома ящик бадвайзера пинта чистого эфира и двенадцать пузырьков амилнитрита не то чтобы все это было категорически необходимо в поездке но если уж начал собирать коллекцию то к делу надо подходить серьезно

Архитектуры

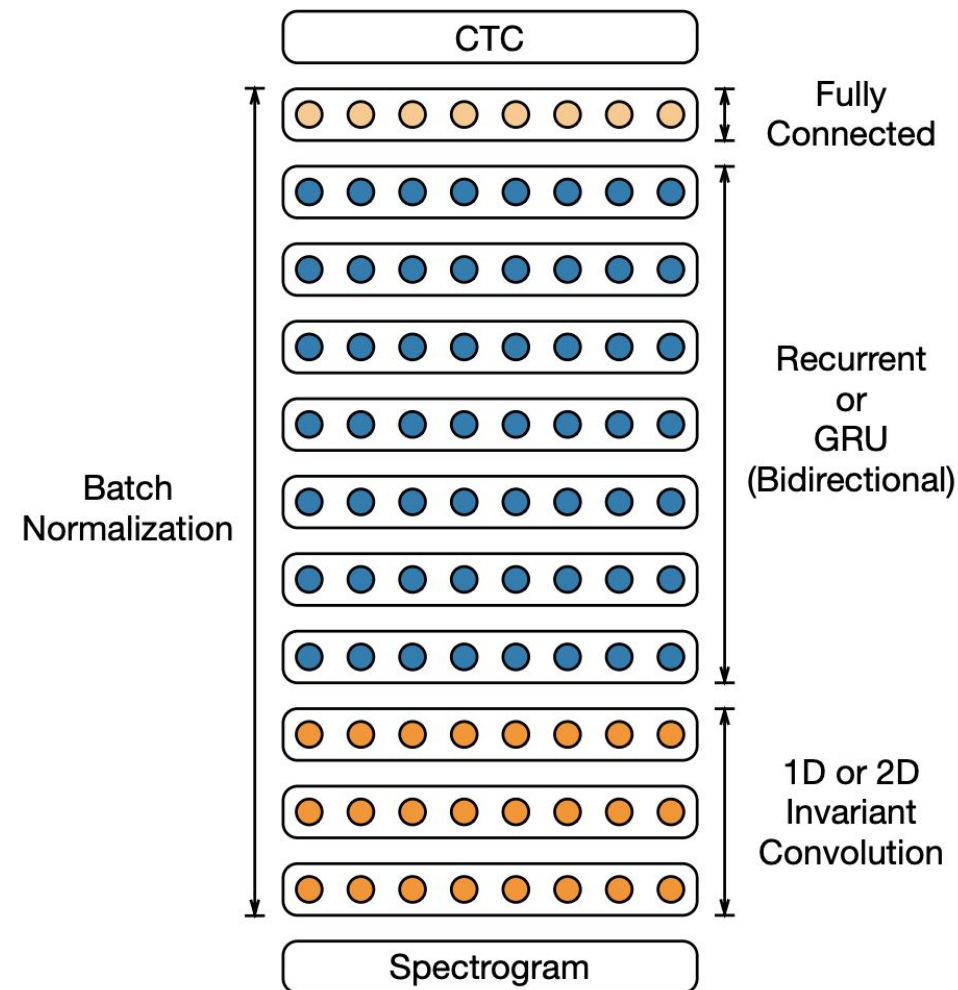
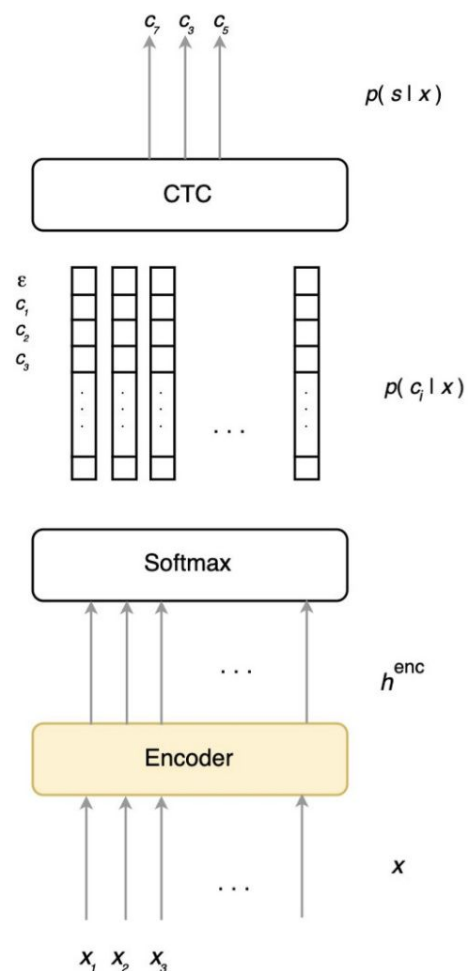
LAS (2015)

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



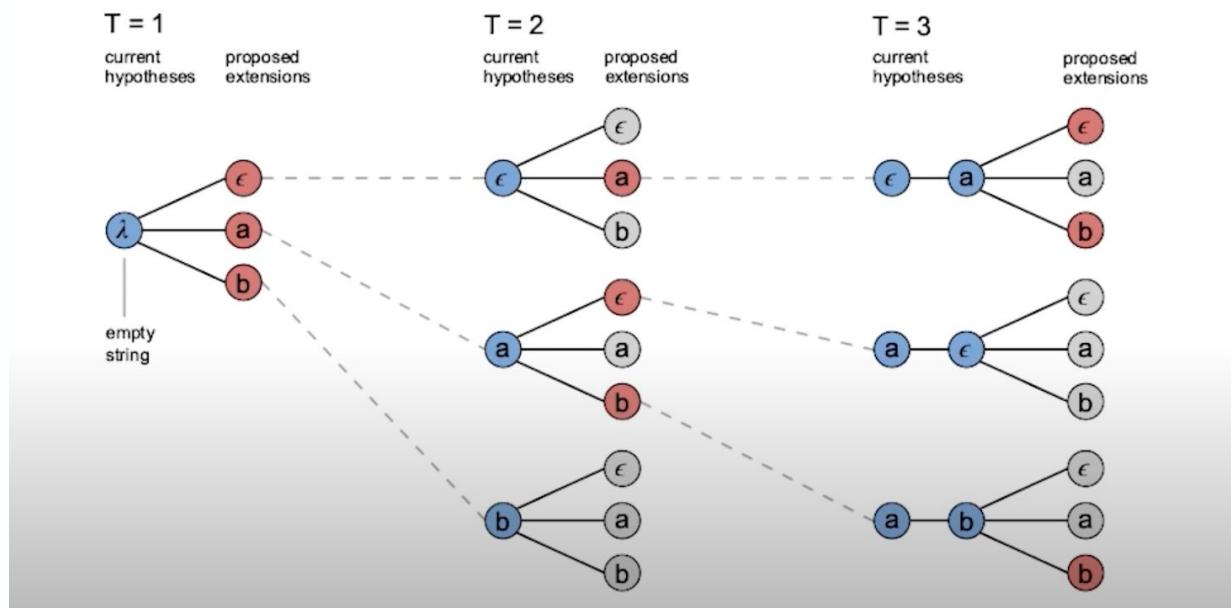
Deep Speech 2 (2015)

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



Beam Search & LM

BEAM SEARCH



Only beam search

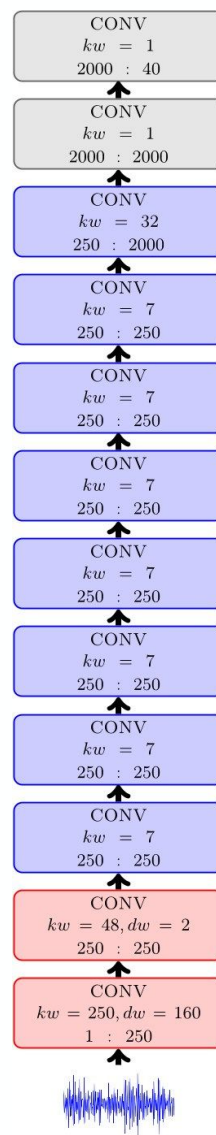
$$\mathbf{y}^* = \arg \max_{\mathbf{y}} \log p(\mathbf{y}|\mathbf{x})$$

Beam search & LM (shallow fusion)

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

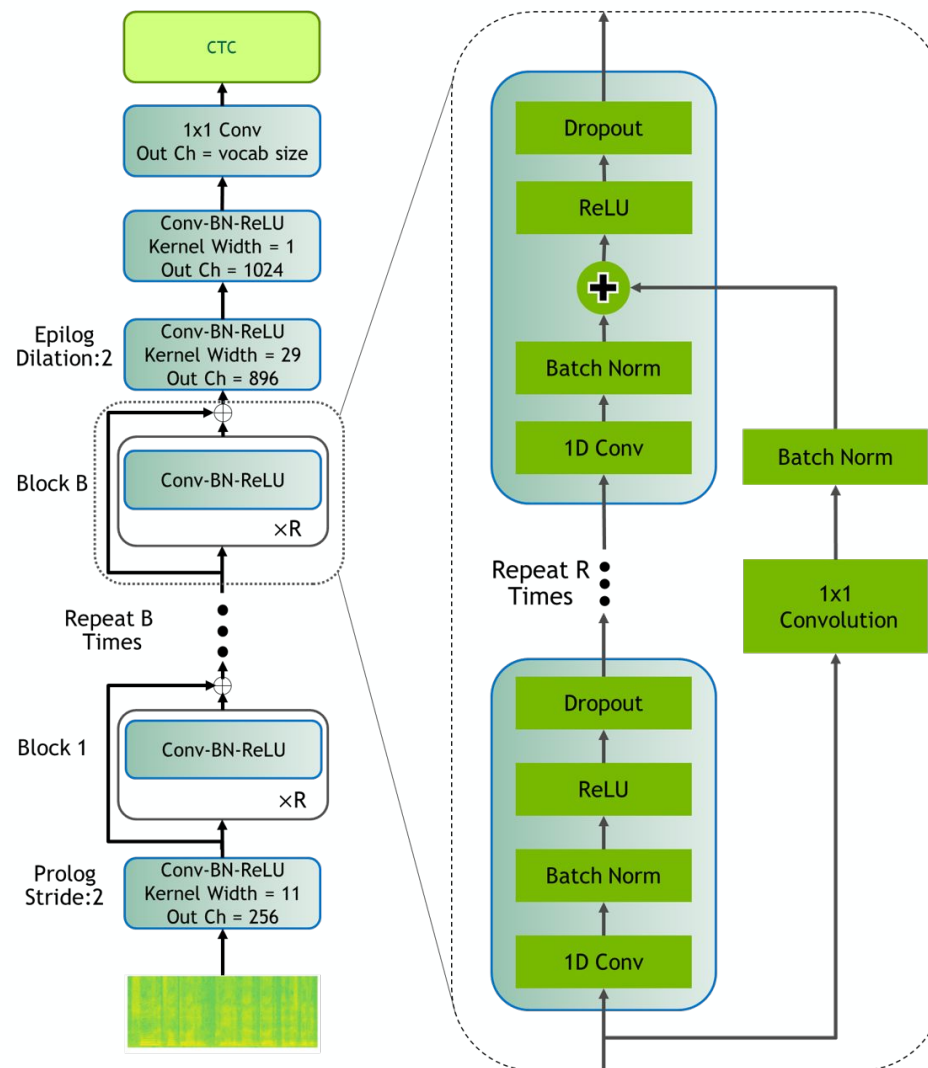
Wav2Letter (2016)

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



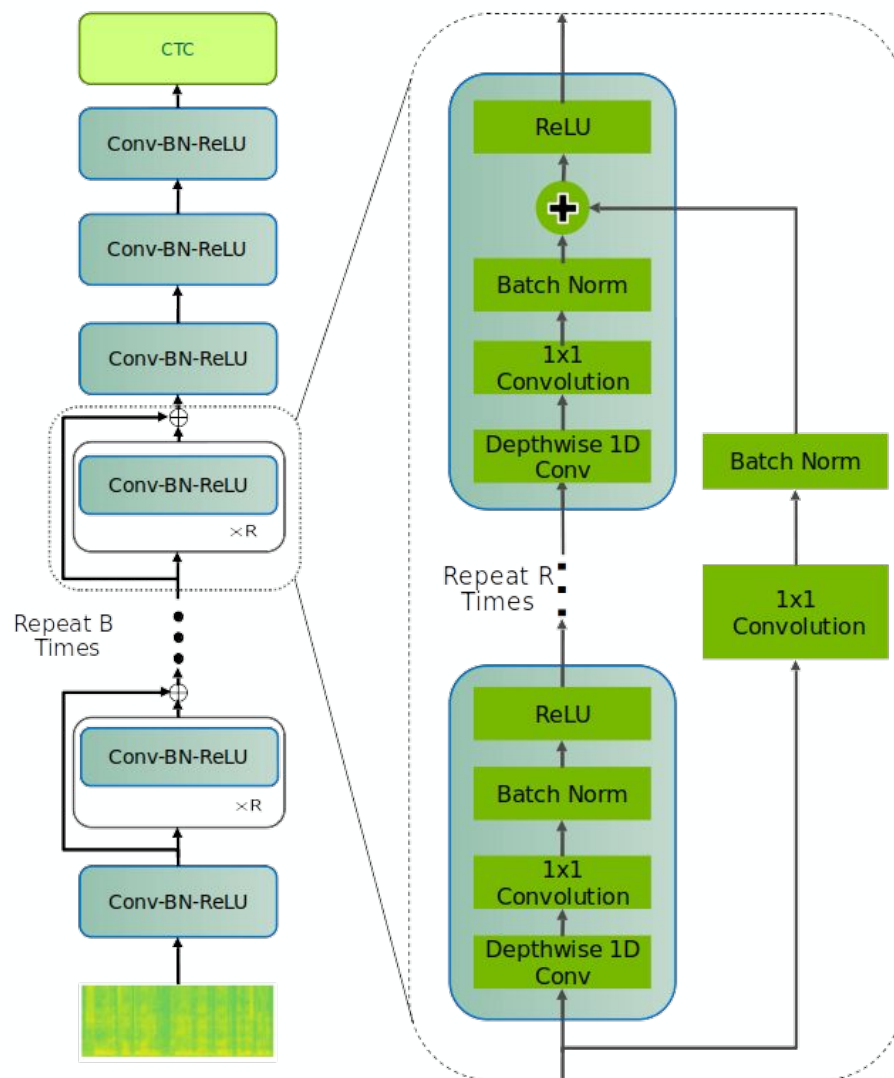
Jasper (2019)

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



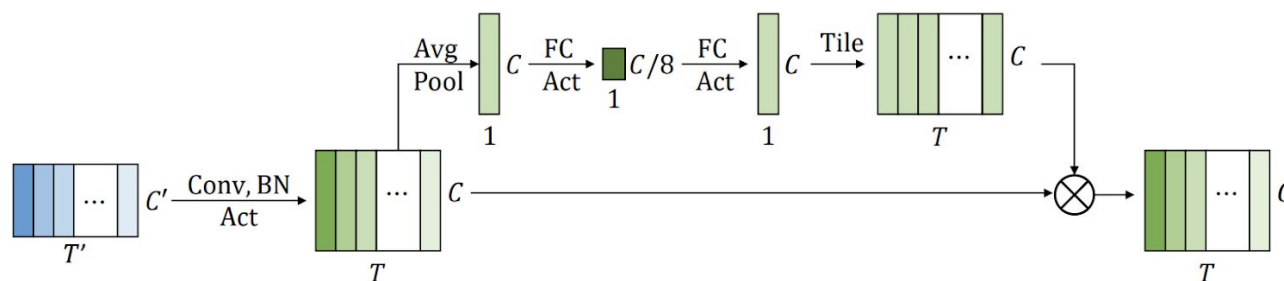
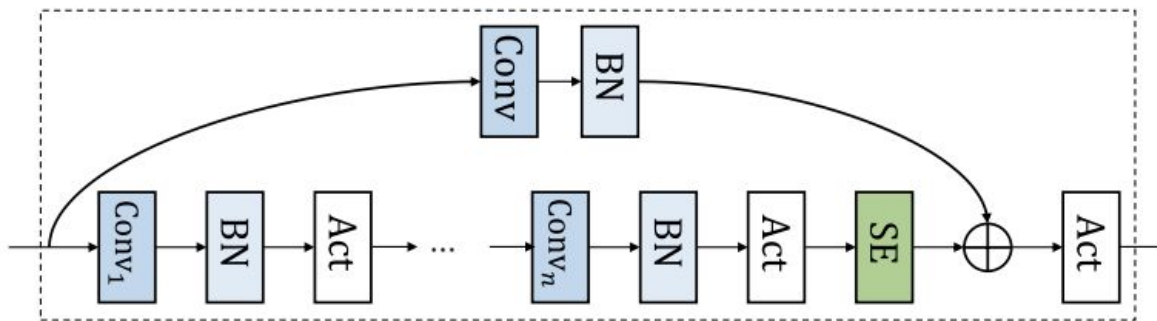
QuartzNet (2019)

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

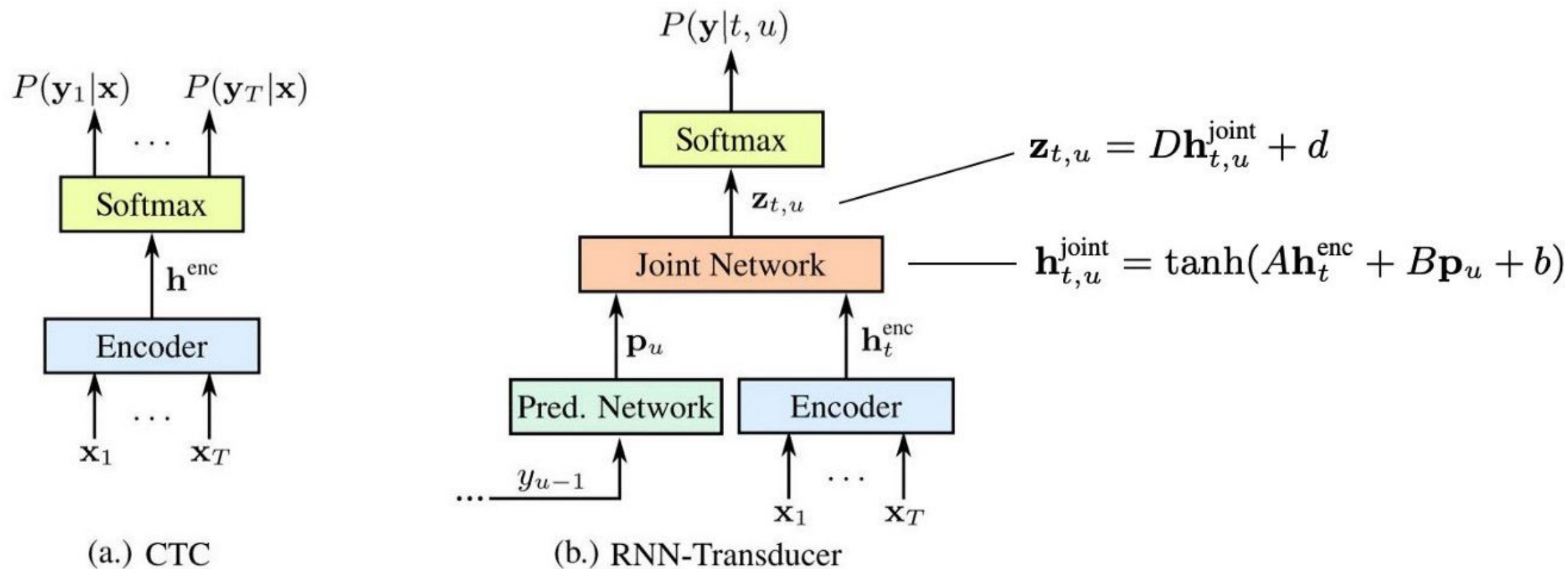


ContextNet (2020)

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



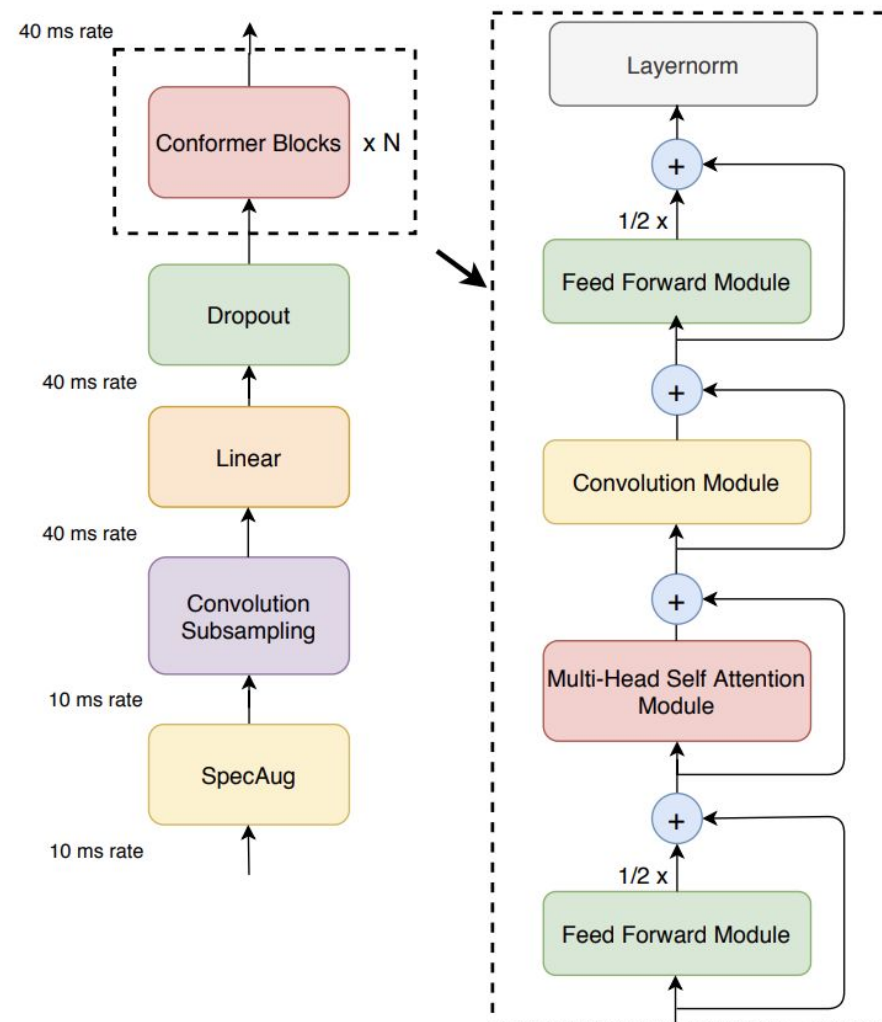
RNN-Transducer



Conformer (2020)

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

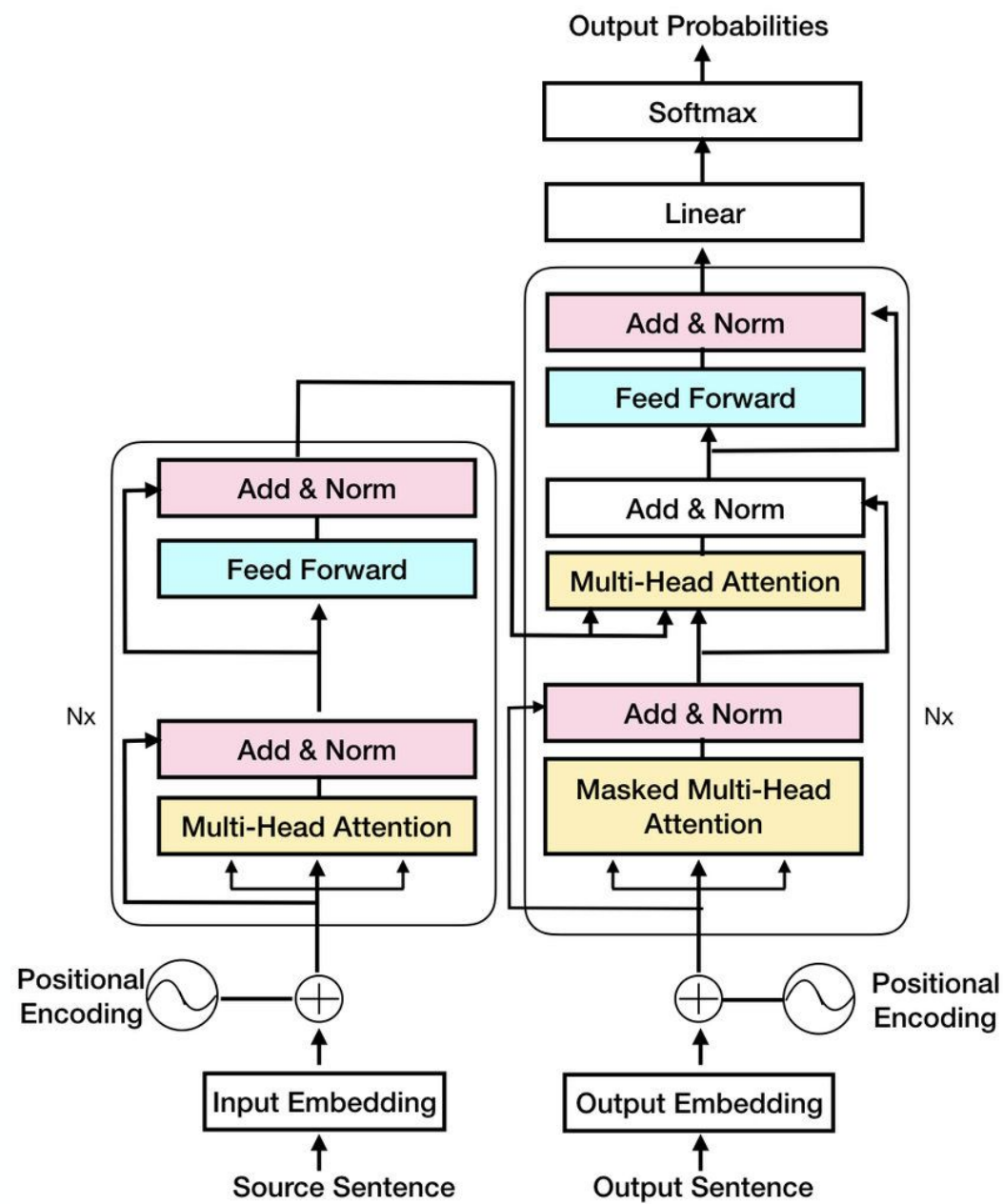
Method	#Params (M)	WER Without LM		WER With LM	
		testclean	testother	testclean	testother
Hybrid					
Transformer [33]	-	-	-	2.26	4.85
CTC					
QuartzNet [9]	19	3.90	11.28	2.69	7.25
LAS					
Transformer [34]	270	2.89	6.98	2.33	5.17
Transformer [19]	-	2.2	5.6	2.6	5.7
LSTM	360	2.6	6.0	2.2	5.2
Transducer					
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

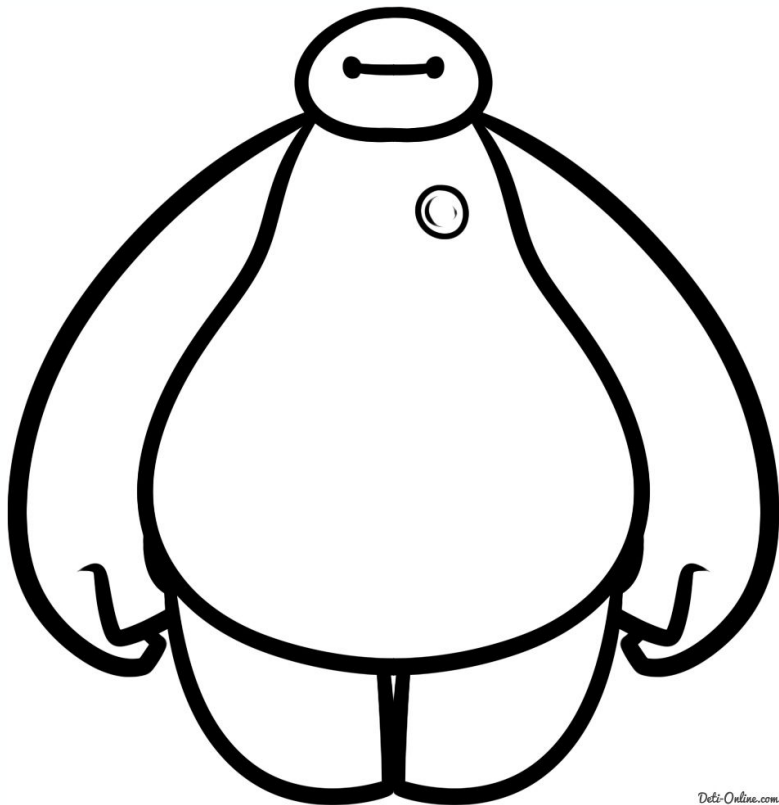


- Препроцессинг
- Акустическая модель
- Языковая модель
- Пунктуационная модель

**привет соня пойдем
сегодня в кинчик**

**Привет, Соня. Пойдем
сегодня в кинчик?**

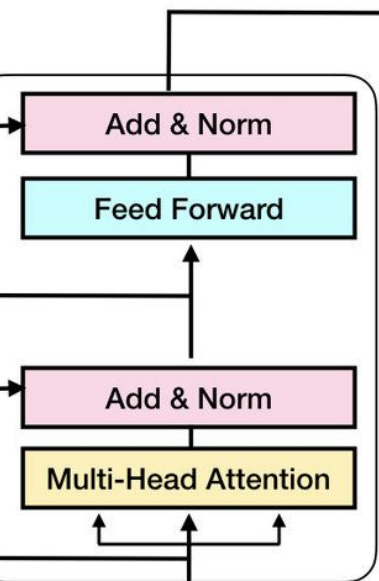




Deti-Online.com

Nx

Positional
Encoding



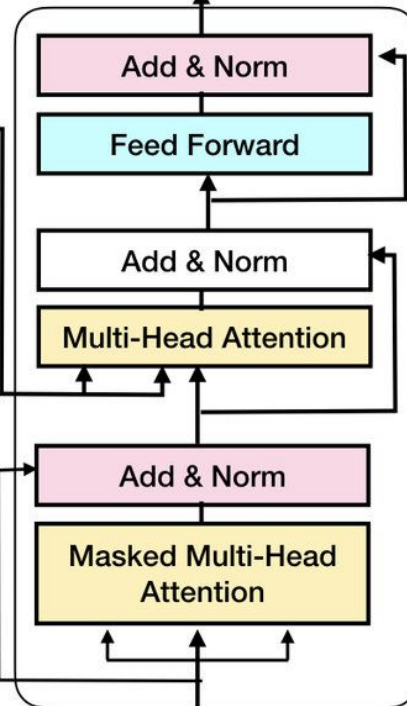
Input Embedding

Source Sentence

Output Probabilities

Softmax

Linear



Nx

Positional
Encoding

Output Embedding

Output Sentence

- -

This paper describes a punctuation restoration system for automatically transcribed Estonian broadcast speech that uses long short-term memory (LSTM)[7]. LSTM, a type of recurrent neural network (RNN), has been used for a variety of supervised sequence labelling tasks, including phoneme classification [8]

This paper describes a punctuation prediction system for automatically transcribed English speech that uses long short-term memory (LSTM), a type of recurrent neural network, which is used for a variety of supervised sequence labeling tasks, including phoneme classification [8]

non-autoregressive?

data

**привет соня пойдём
сегодня в кинчик**

data

Привет, Соня!
Пойдем сегодня в кинчик?

data

_привет	1,	_Привет,
_со	1	_Со
_ня	0!	_ня!
_пойдем	1	_Пойдем
_сегодня	0	_сегодня
_в	0	_в
_кин	0	_кин
_чик	0?	_чик?

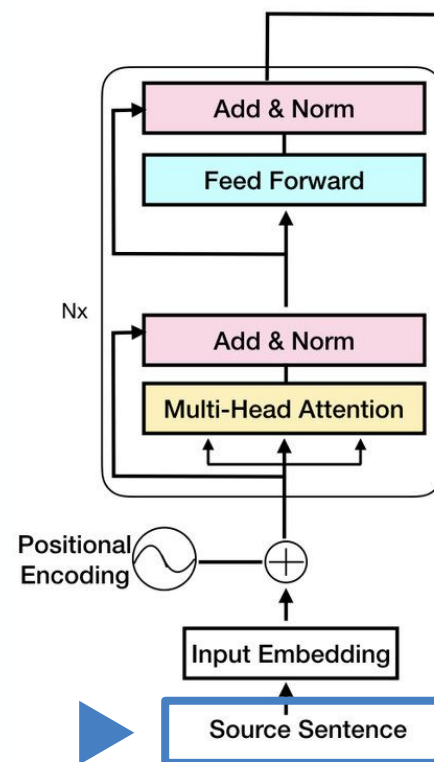
punctuation& capitalization

привет соня пойдем
сегодня в кинчик

youtokentome

pre-trained embeddings

https://www.researchgate.net/publication/338223294_Neural_Machine_Translation_for_the_Bangla-English_Language_Pair/figures?lo=1



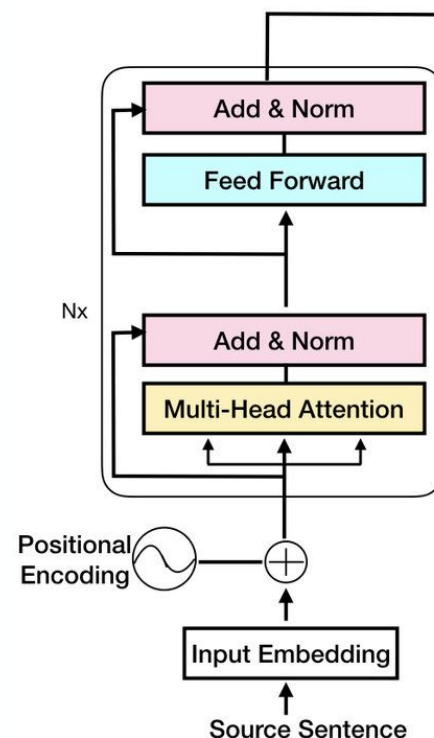
```
self.out =  
nn.Sequential(  
  
nn.Linear(self.encoder.h_s,  
self.encoder.h_s),  
GELU(),  
  
nn.Linear(self.encoder.  
h_s, n_classes),  
)
```

punctuation& capitalization

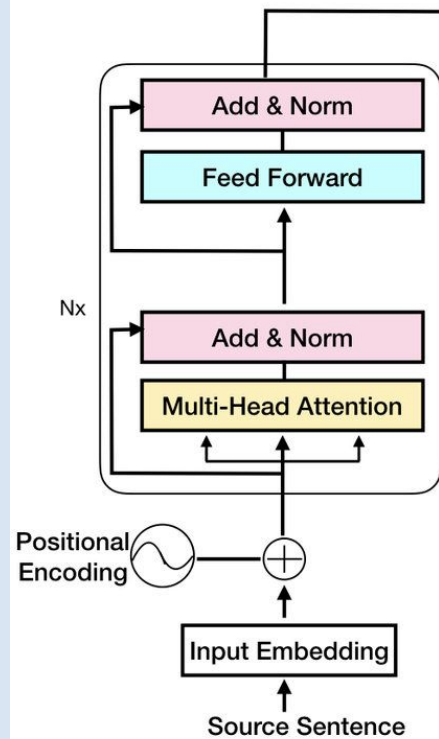
1,10!10000?

ch2idx

https://www.researchgate.net/publication/338223294_Neural_Machine_Translation_for_the_Bangla-English_Language_Pair/figures?lo=1



```
self.out =  
nn.Sequential(  
  
nn.Linear(self.encoder.h_s,  
self.encoder.h_s),  
GELU(),  
  
nn.Linear(self.encoder.  
h_s, n_classes),  
)
```



```
self.out =
nn.Sequential(

nn.Linear(self.en
coder.h_s,
self.encoder.h_s)

,

GELU(),

nn.Linear(self.en
coder.
h_s, n_classes),

)
```

