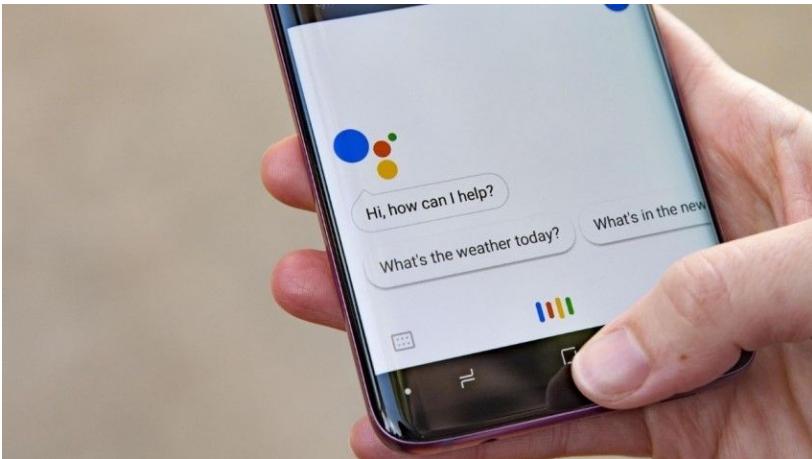


Automatic Speech Recognition

Andrusenko Andrei
Research scientist at STC

1. Область применения Automatic Speech Recognition (ASR).
2. Постановка задачи, критерий оценки качества.
3. Краткий экскурс в историю развития ASR.
4. Как устроена ASR система?
 - a. DTW.
 - b. Hybrid.
 - c. End-to-end.
5. Сравнение hybrid-based и end-to-end систем.
6. Примеры реальных кейсов компаний.
7. Где обучать ASR системы?

Голосовые ассистенты



Умные колонки



Про сравнения с «Яндексом», нужды людей и захват рынка ассистентов: рассказывает конструктор техники от «Сбера» ✓

Большое интервью с Константином Кругловым – он разработал колонки с «Алисой» и теперь проектирует будущее голосовых помощников, рассуждает про рынок и объясняет, когда на голосовых запросах наконец-то можно будет заработать.

+12 Q 91 комментарий

В закладки

Слушать



Технологии умного дома



Речевая аналитика



ПРИМЕР РАБОТЫ:



Толщина
левой доли 70 мм.
Контуры чёткие...

распознавание речи



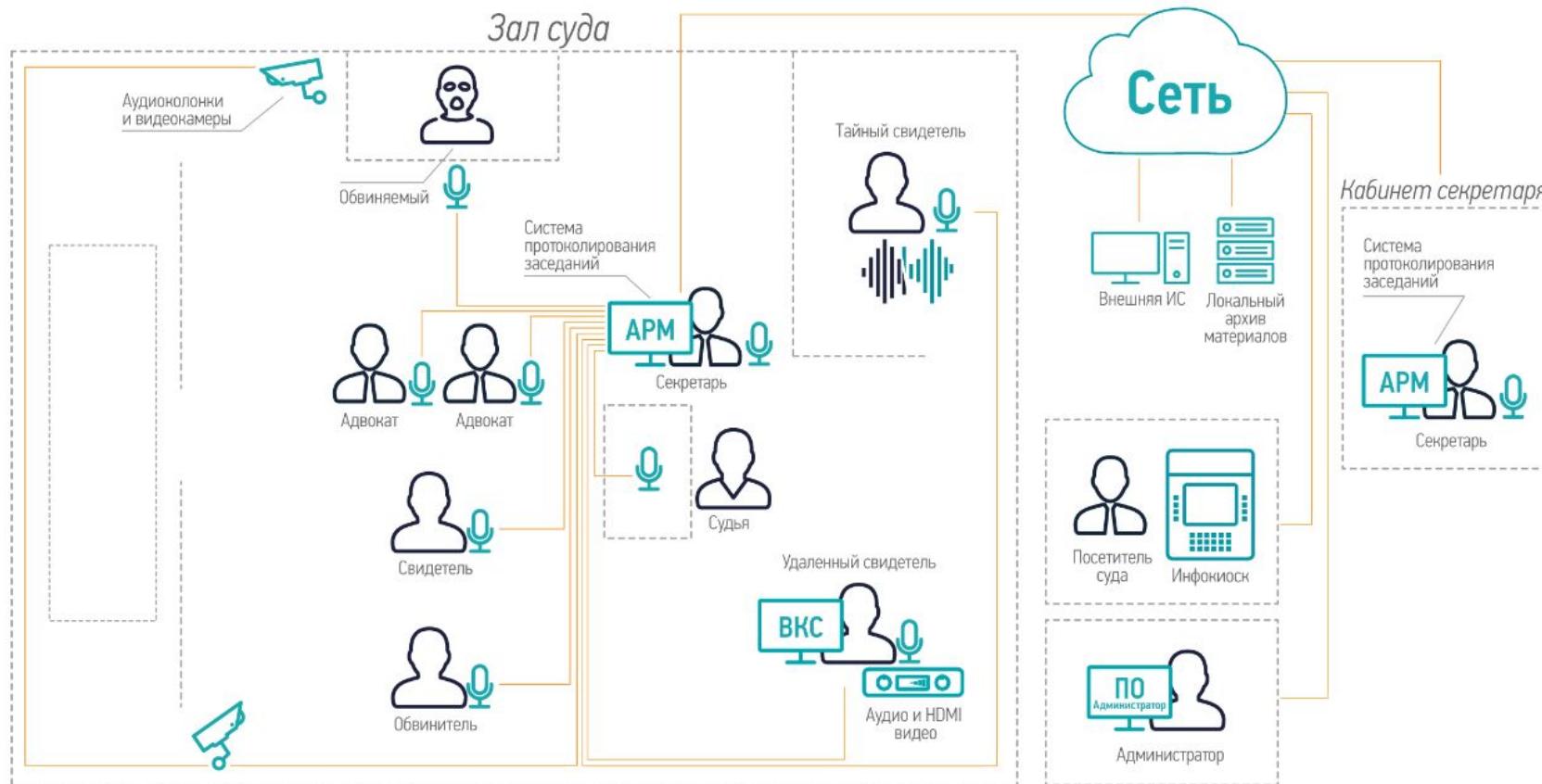
заполненный
протокол



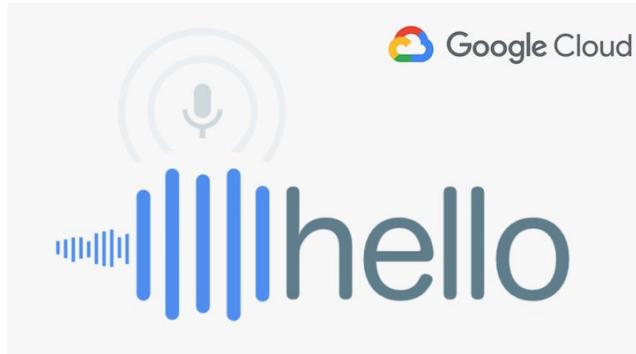
Протоколирование мероприятий



Протоколирование мероприятий



Облачные платформы



<https://cloud.google.com/speech-to-text>



Yandex SpeechKit

<https://cloud.yandex.ru/services/speechkit>



<https://cp.speechpro.com/service/asr>



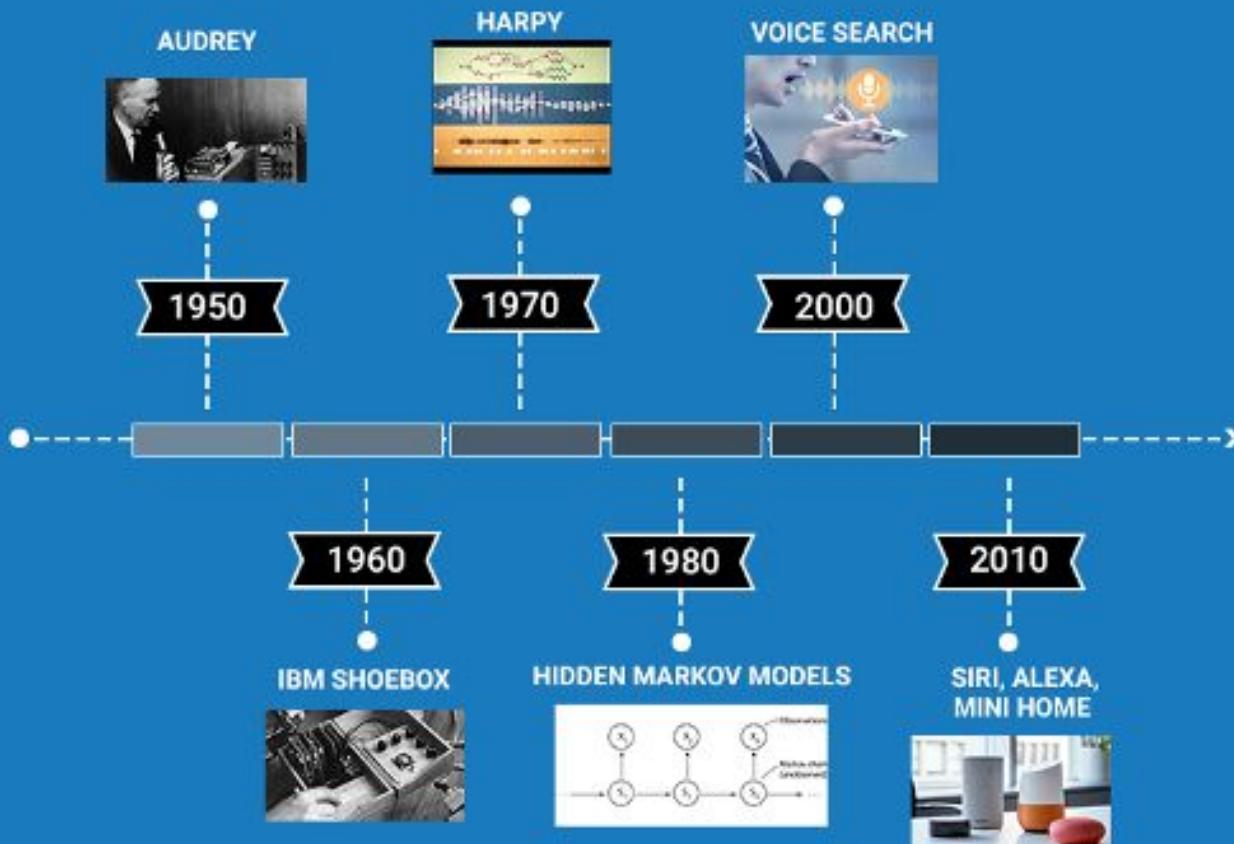
- ▶ Выравнивание (по Левенштейну):
 - ▶ Эталон: Мой **дядя**, самых **честных** правил, когда **не** в шутку **занемог**...
 - ▶ Распознано: Мой **дятер** самых **честь не** правил когда в шутку **за не мог**
 - ▶ Замены (substitutions)
 - ▶ Вставки (insertions)
 - ▶ Удаления (deletions)
- ▶ WER (Word Error Rate) – пословная ошибка распознавания
- ▶ Accuracy – точность распознавания

$$WER = \frac{\#замен + \#вставок + \#удалений}{\#\text{слов в эталоне}} * 100\%$$

$$Accuracy = 100\% - WER$$

С чего все началось?







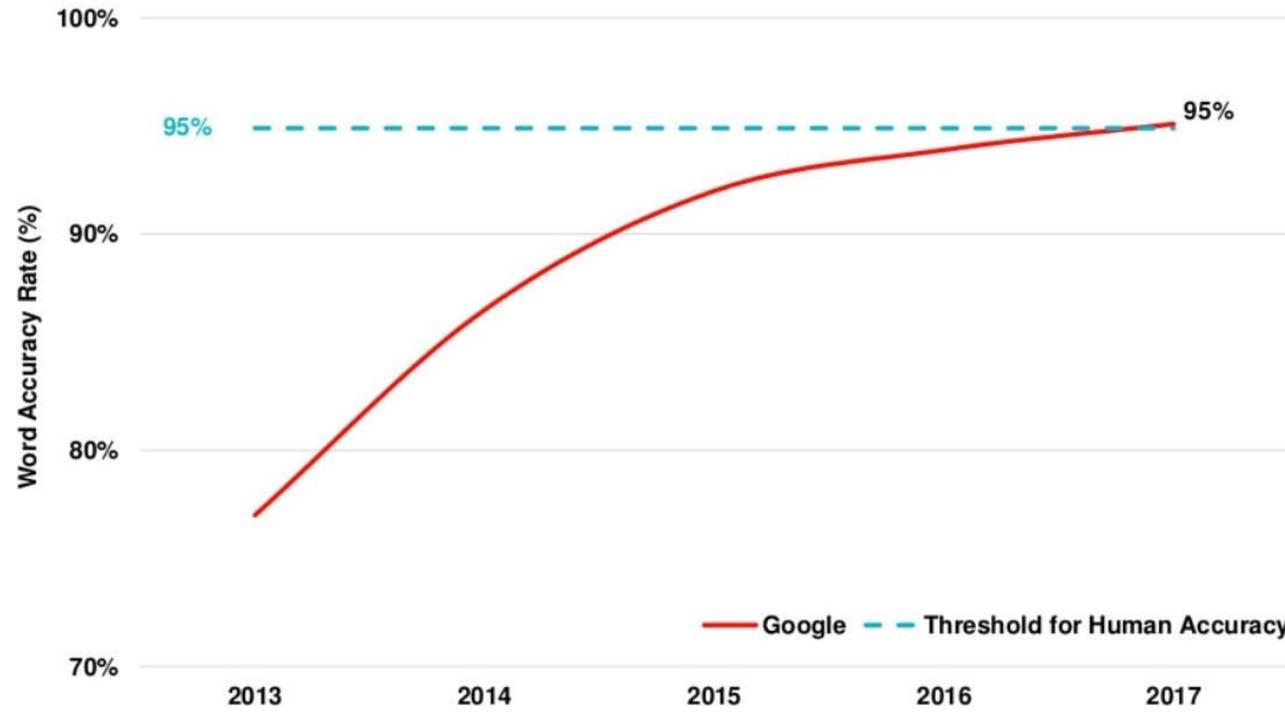
- В 1952, лаборатория Bell изобрела систему “Audrey”, которая была способна распознавать произнесения цифр.
- Десять лет спустя, IBM представила систему “Shoebox”, которая могла понимать 16 английских слов.
- 1970 – Carnegie Mellon’s “Harpy” голосовая система была способна понимать уже более 1,000 слов, что схоже с вокабуляром трехлетнего ребенка..

- Использование в 80-х статистических методов, таких как HMM (Hidden Markov Model) позволило увеличить словарь систем распознавания до нескольких тысяч.
- В 2000-х, системы распознавания речи достигли 80% точности в задачах диктовки. В Google появляется первый голосовой помощник для поиска.
- В 2011 Apple анонсировали первый релиз Siri.

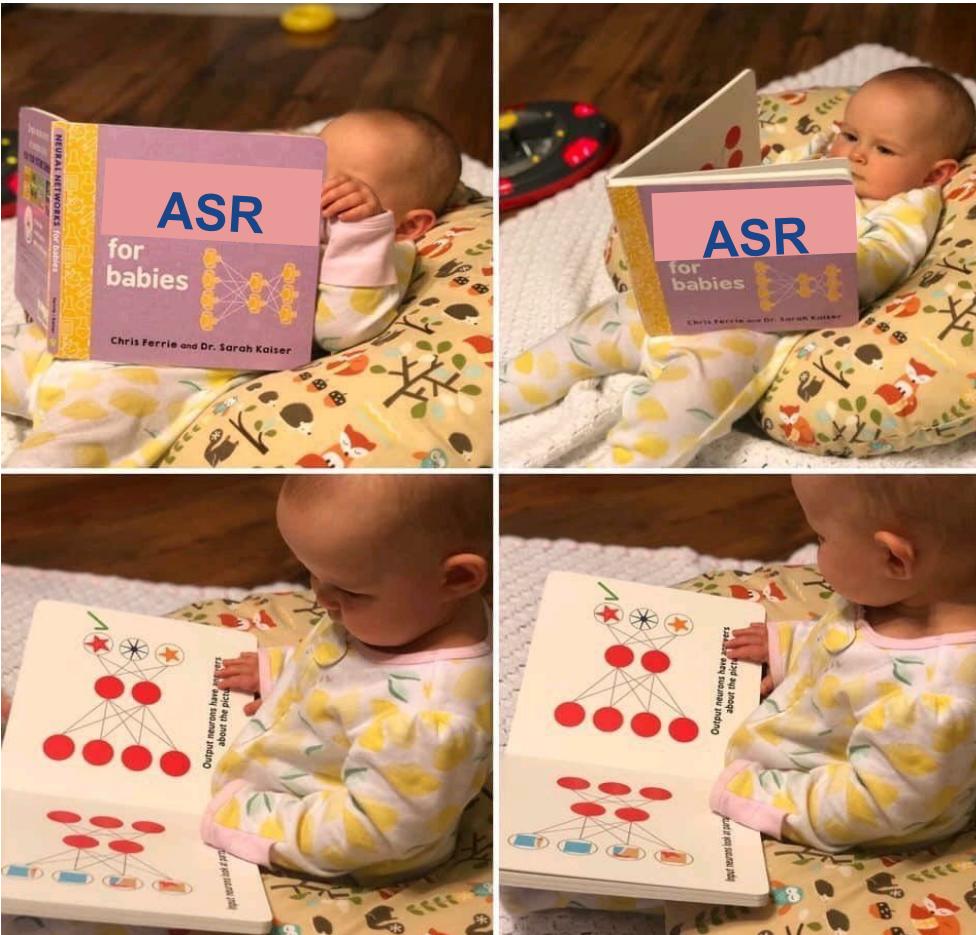


Google Machine Learning

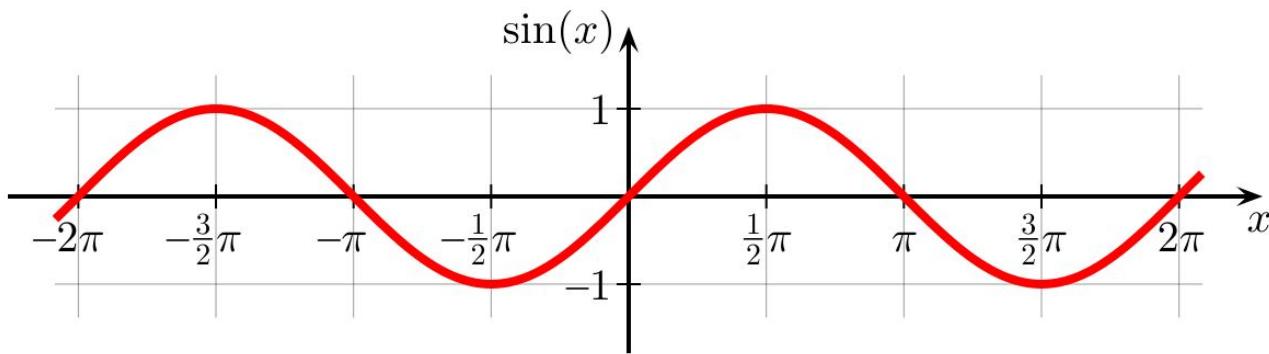
Achieving Higher Word Accuracy, 2013-2017



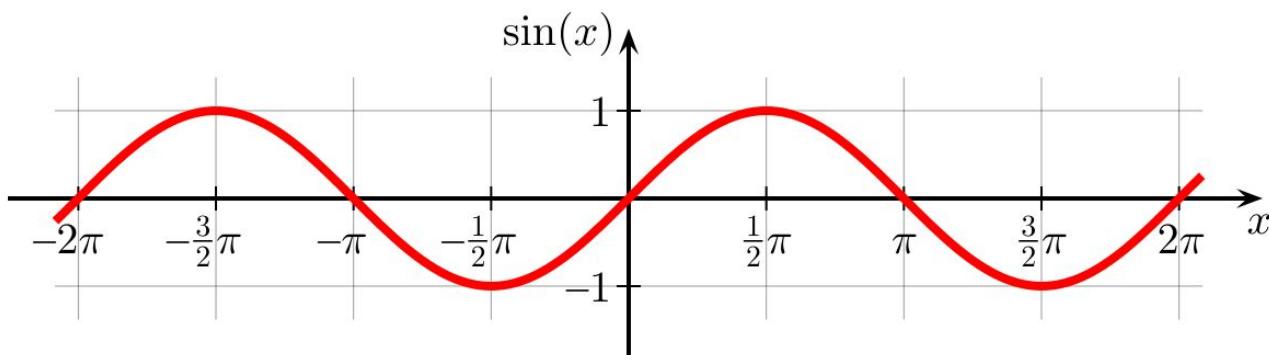
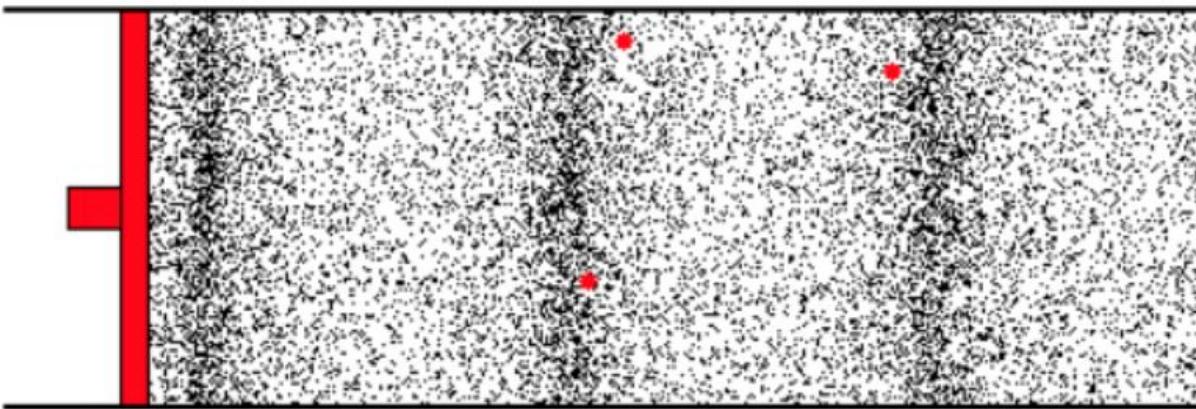
Как все устроено?

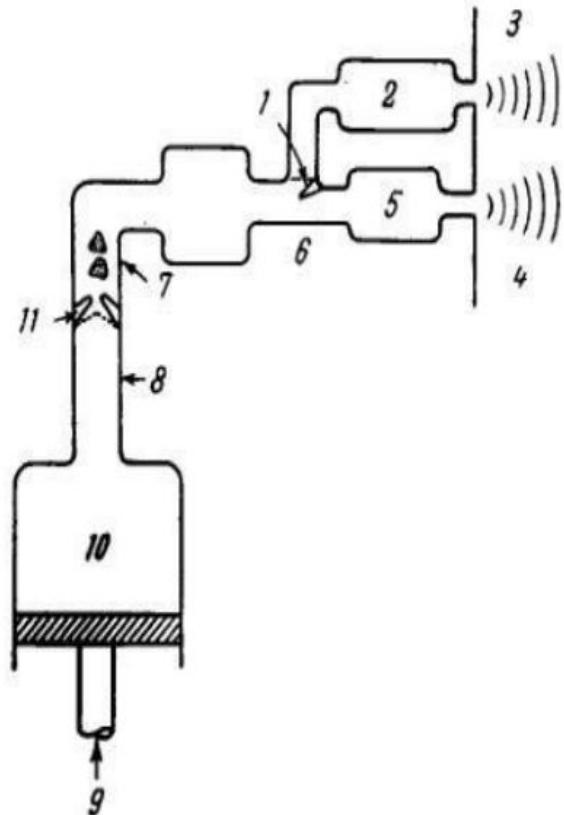


Что такое звук?



Longitudinal Wave





1 — небная занавеска, 2 — носовая полость, 3 — излучения носового тракта, 4 — излучения рта, 5 — ротовая полость, 6 — подножная часть языка, 7 — гортанная трубка, 8 — трахея и бронхи, 9 — мускульная сила, 10 — объем легких, 11 — голосовые связки

Представление речевого сигнала

Step 1: Analog audio signal - Continuous representation of signal



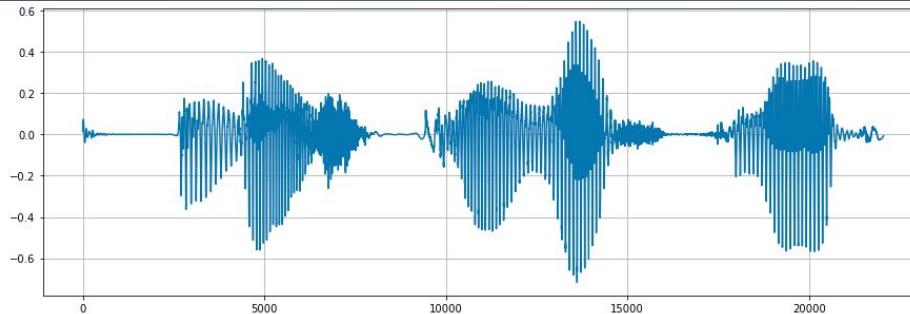
Step 2: Sampling - Samples are selected at regular time intervals



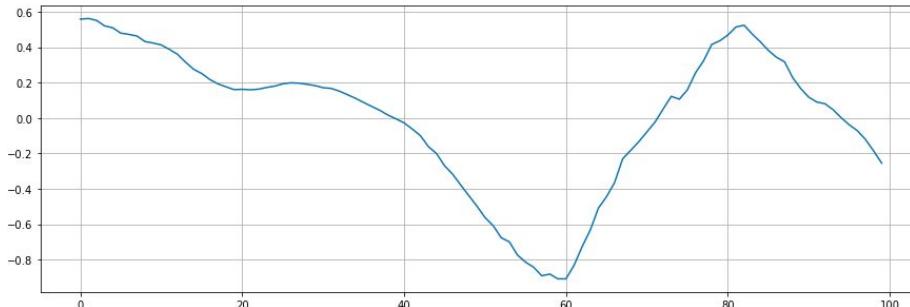
Step 3: Digital audio signal - The way it is stored in memory



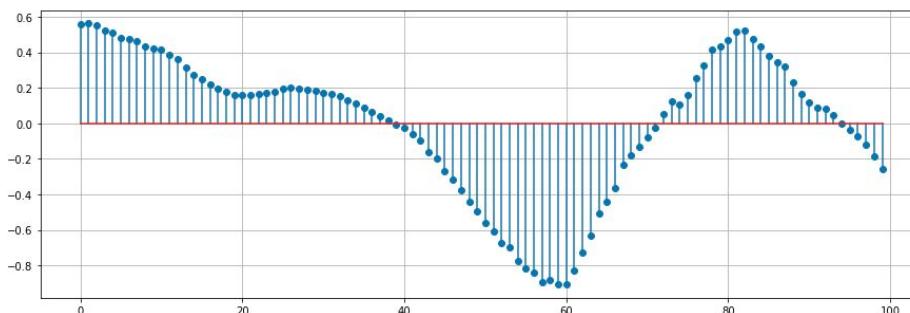
Представление речевого сигнала



```
import matplotlib.pyplot as plt
plt.plot(x[sr:sr*2])
plt.grid()
plt.show()
```

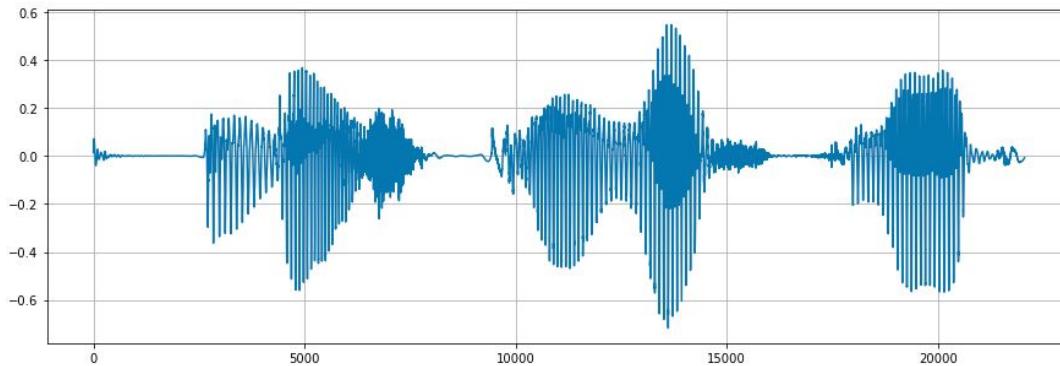


```
plt.plot(x[10900:11000])
plt.grid()
plt.show()
```

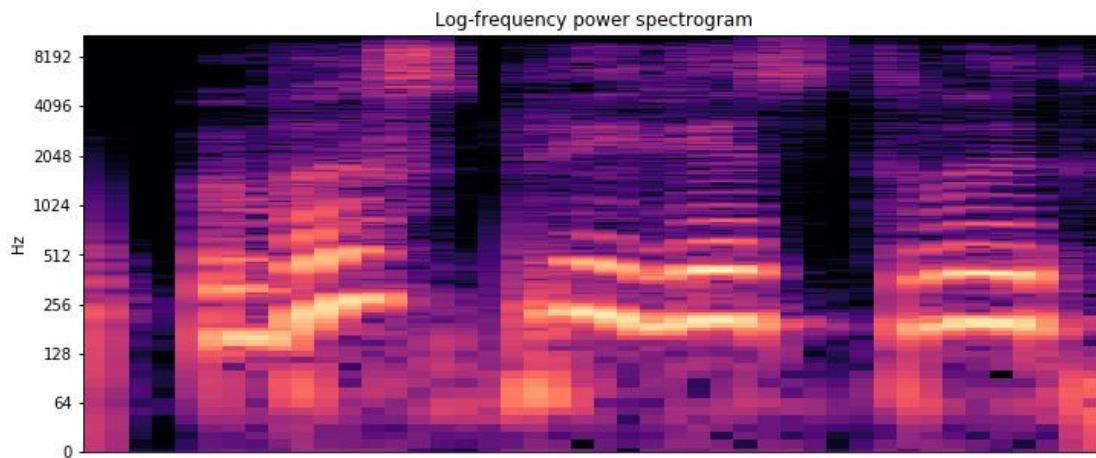


```
plt.stem(x[10900:11000])
plt.grid()
plt.show()
```

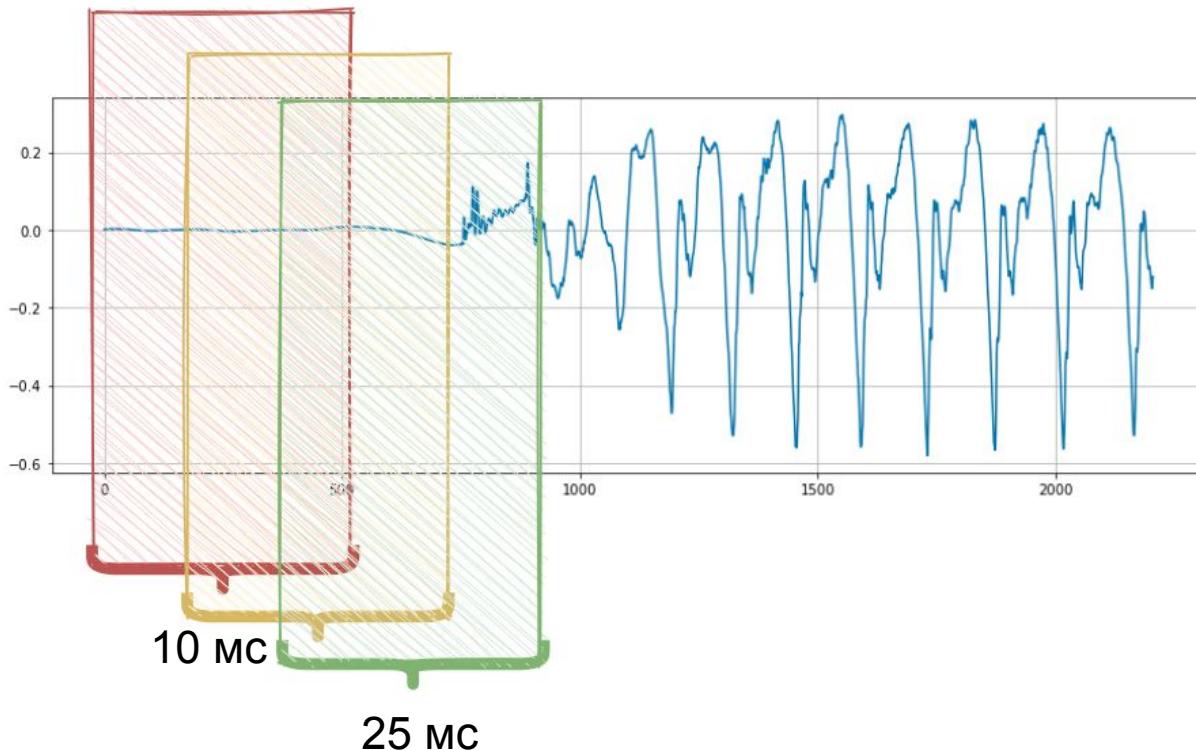
Представление речевого сигнала



```
plt.plot(x[sr:int(sr*2)])
plt.grid()
plt.show()
```

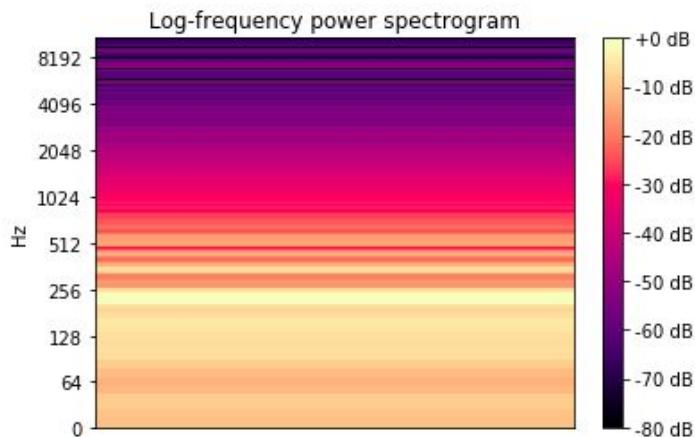
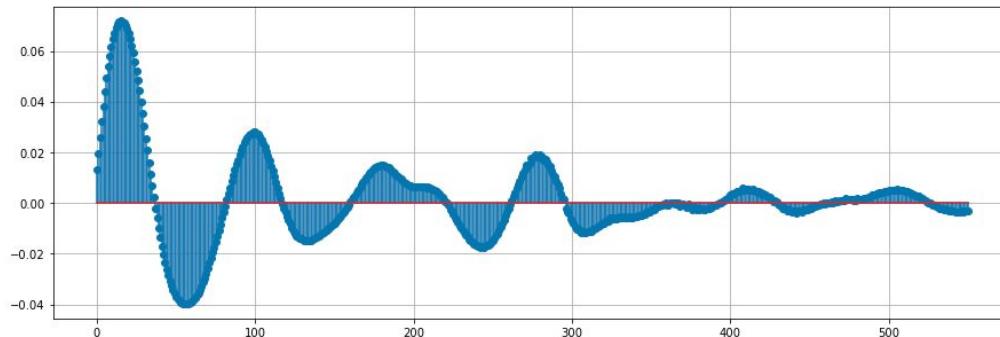


Представление речевого сигнала

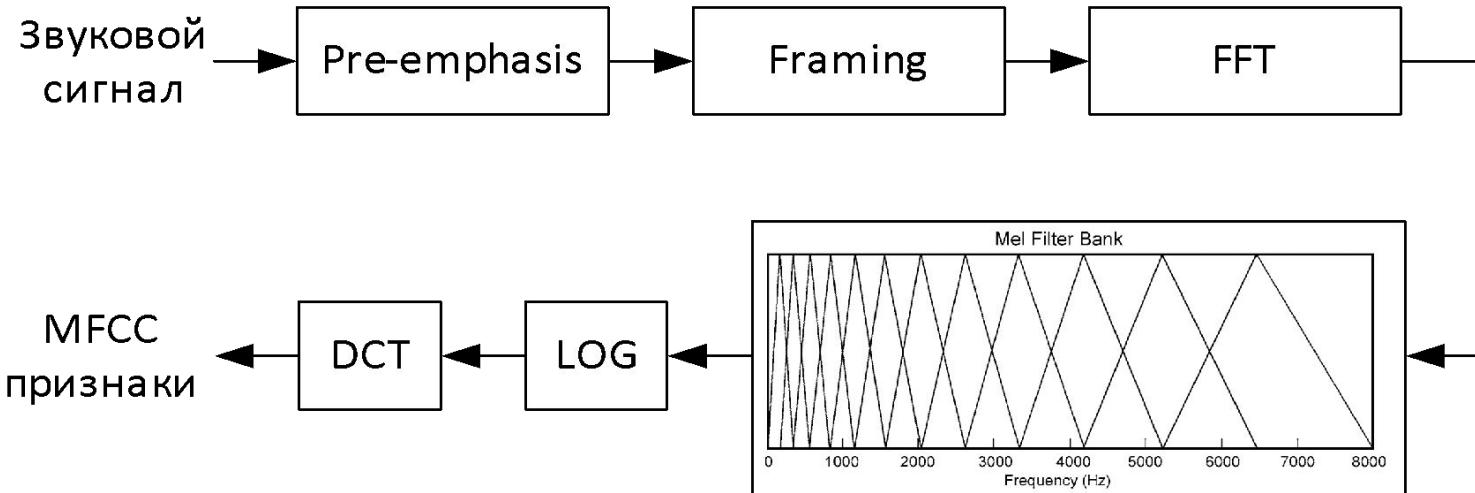
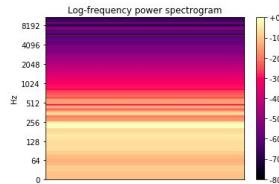
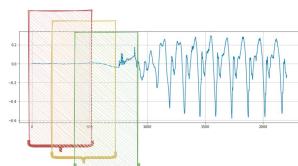
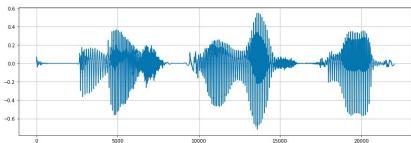


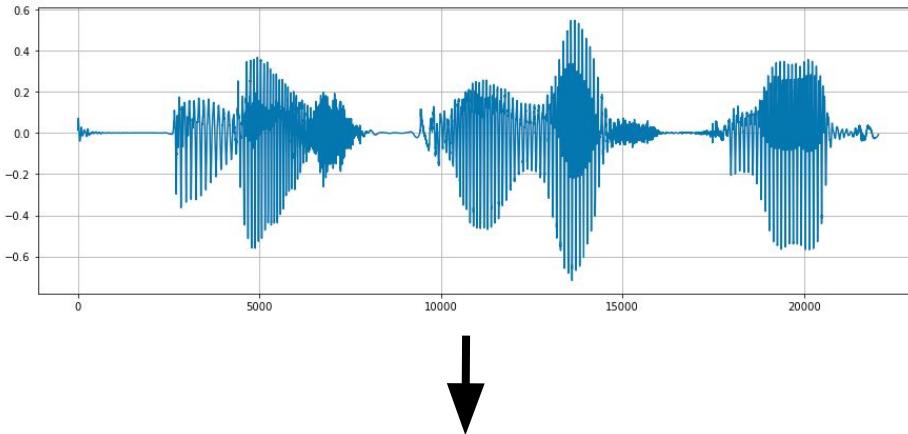
Представление речевого сигнала

```
plt.stem(x[sr:int(sr+sr*0.025)])
plt.grid()
plt.show()
```



Feature extraction



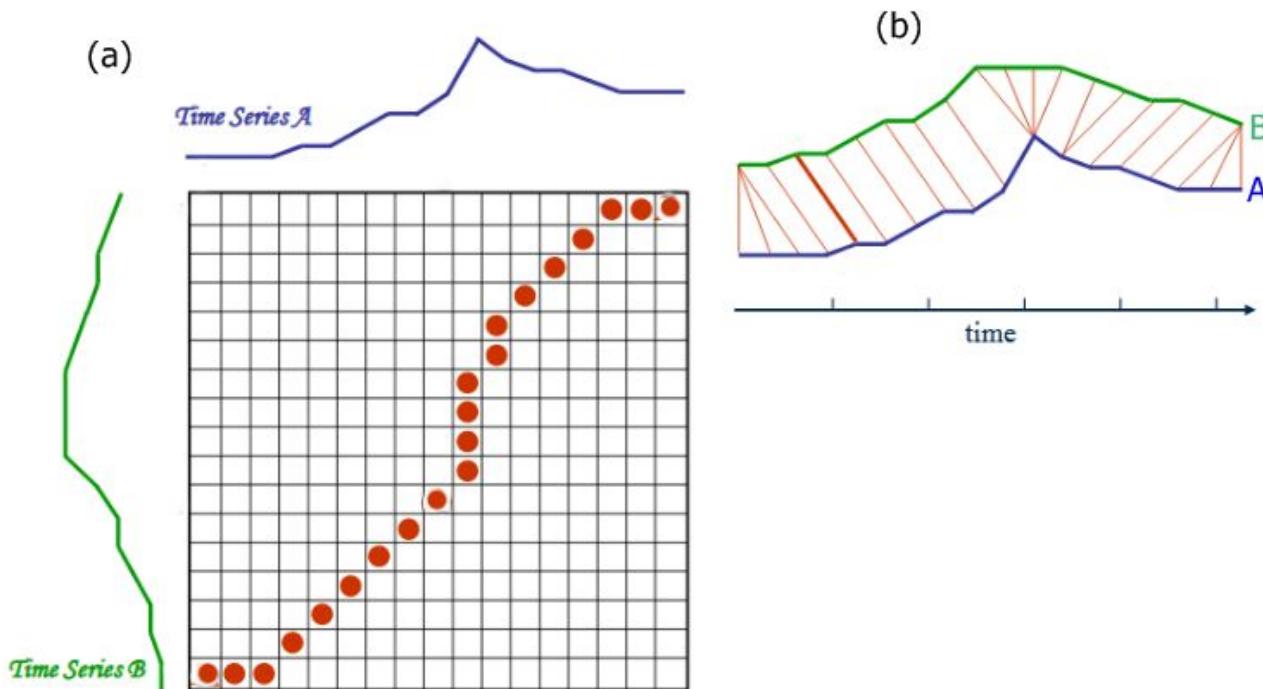


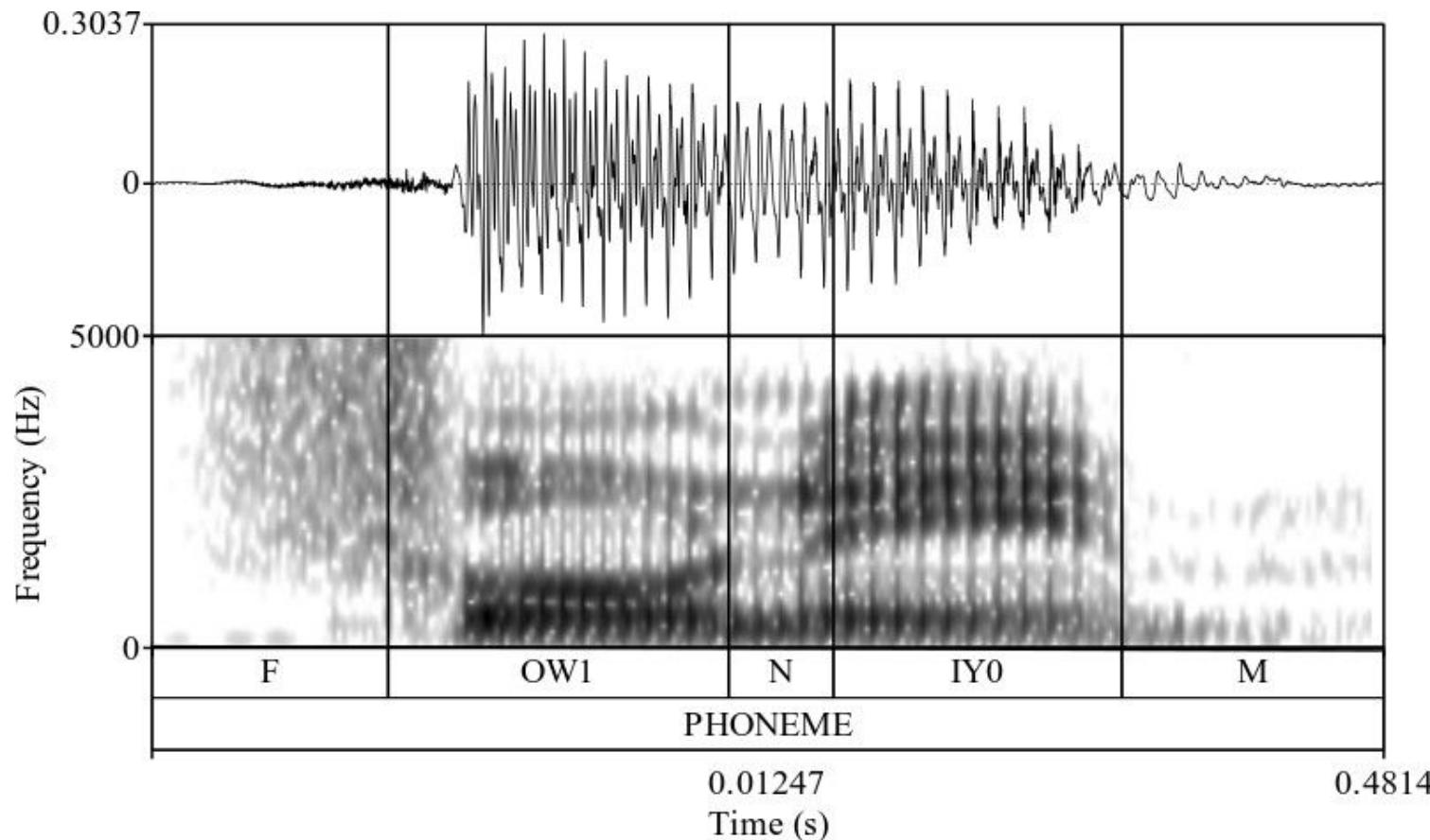
$$X = [x_1, x_2, x_3, \dots, x_n]$$

где x_i – вектор признаков (FBANK, MFCC, ...)

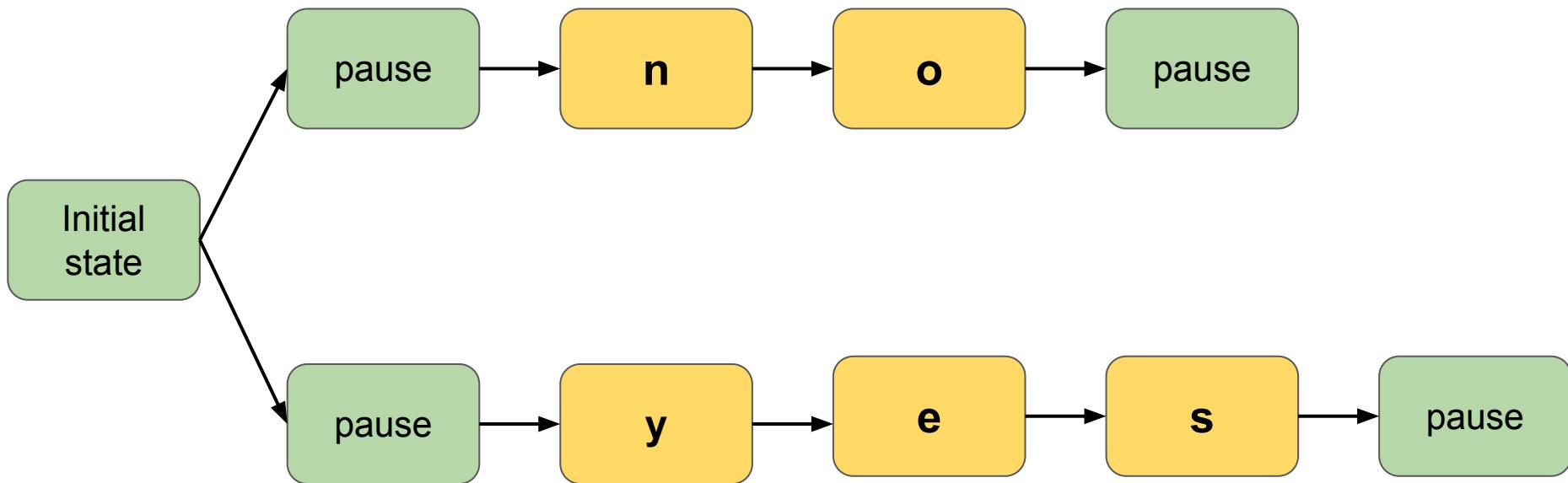
Один кадр (0.025 мс при 16 МГц):

- сырой сигнал = $16000 * 0.025 = 400$ отсчетов амплитуды
- FBANK = 80 отсчетов
- MFCC = 40 отсчетов

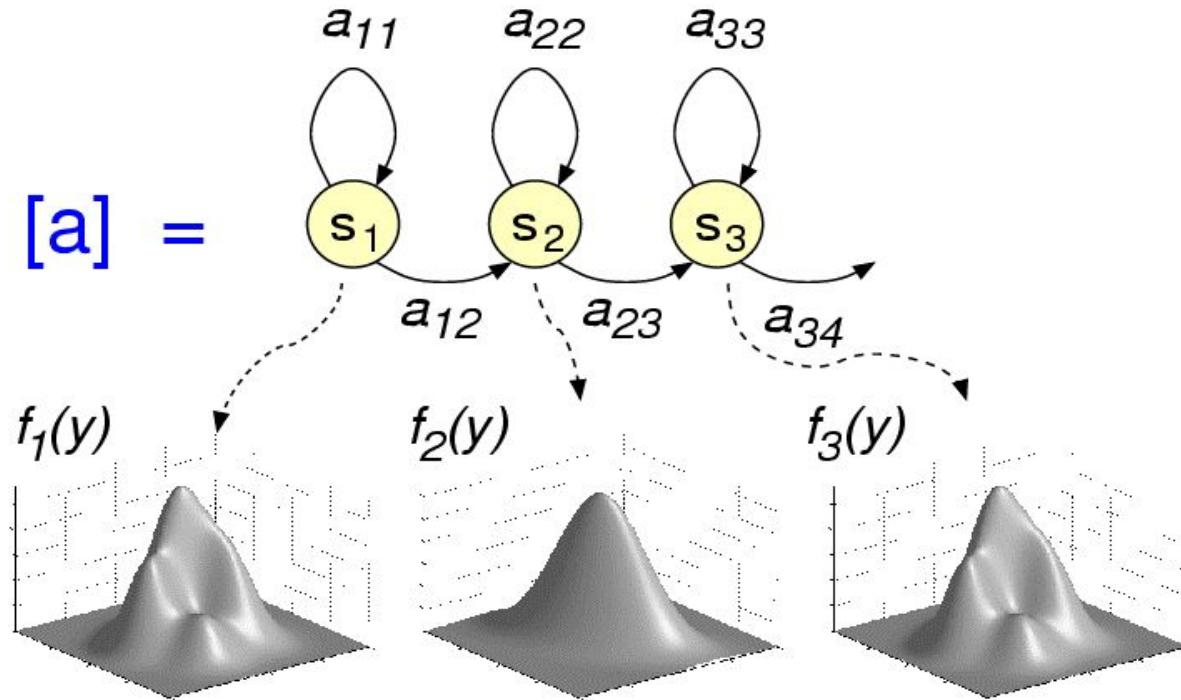




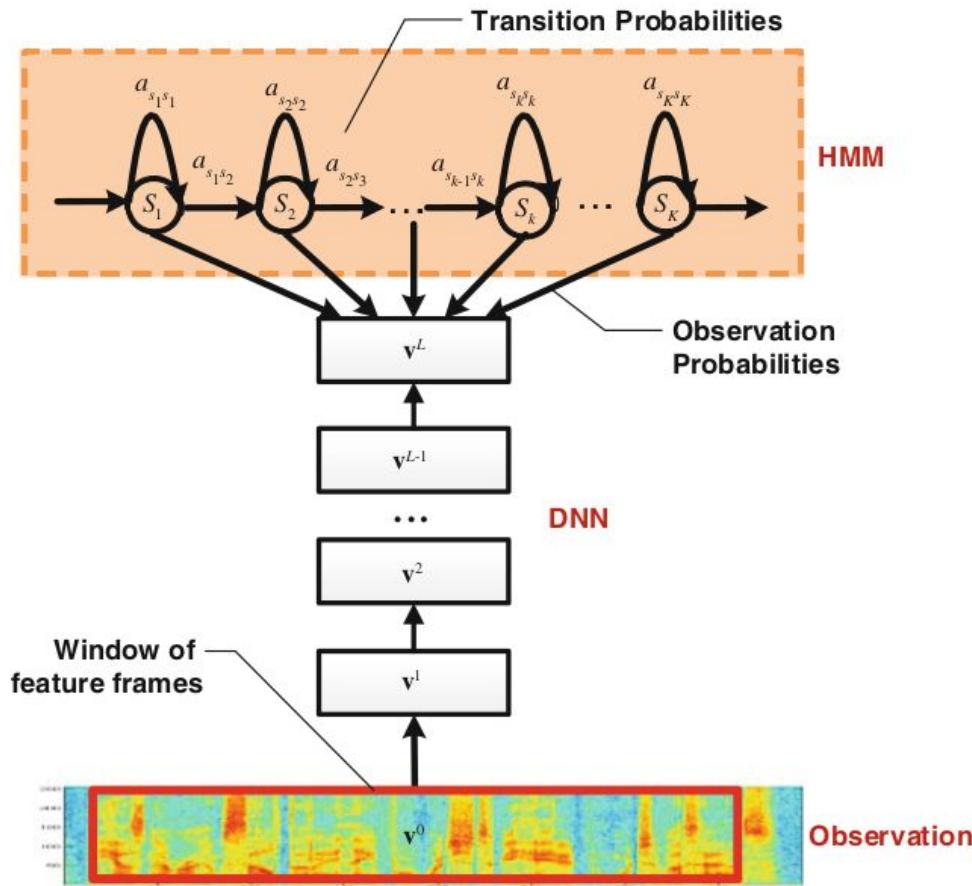
Token passing algorithm



HMM + Gaussian Mixture Models

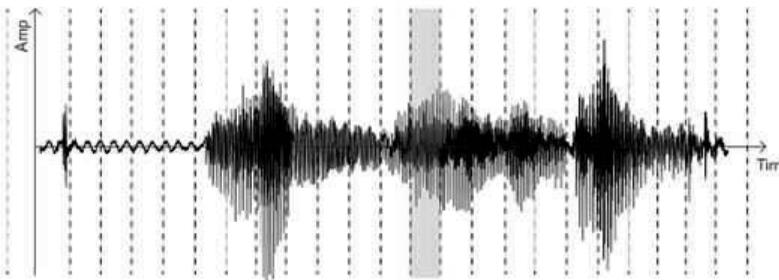


HMM + Deep Neural Network (DNN)

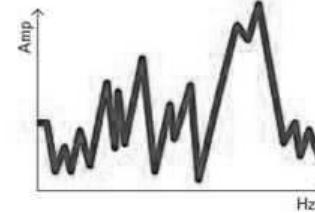


Acoustic model

1 Фреймы



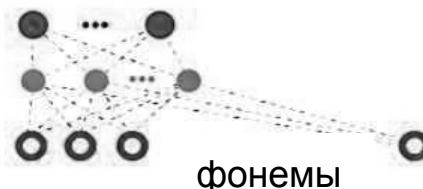
2 Спектр фрейма



3 MFCC

(9, 55.54, 8.113, 3.5553, -1.583, , 2.4, 6.105)

4 Нейронная сеть

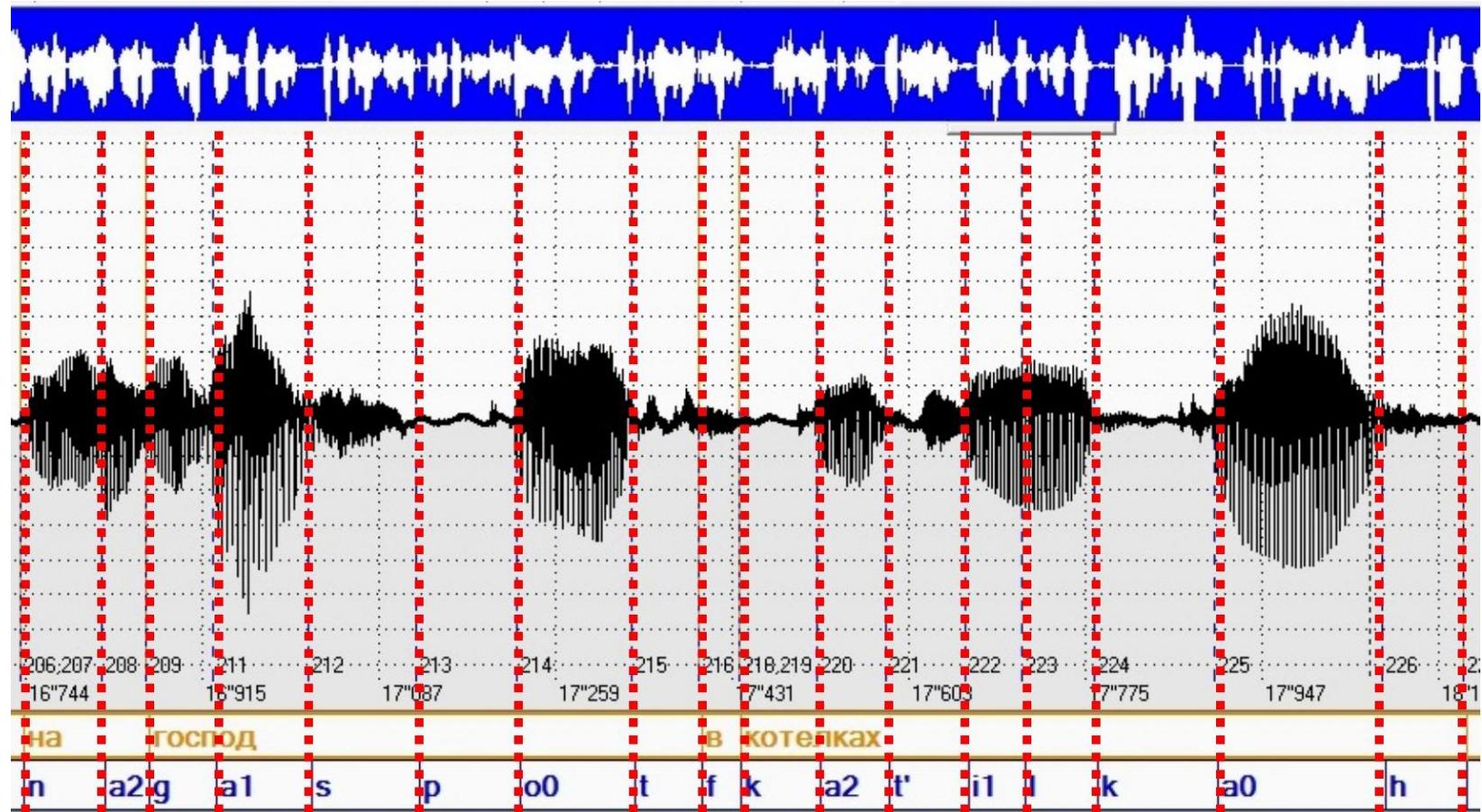


5 Распределение вероятностей по фонемам

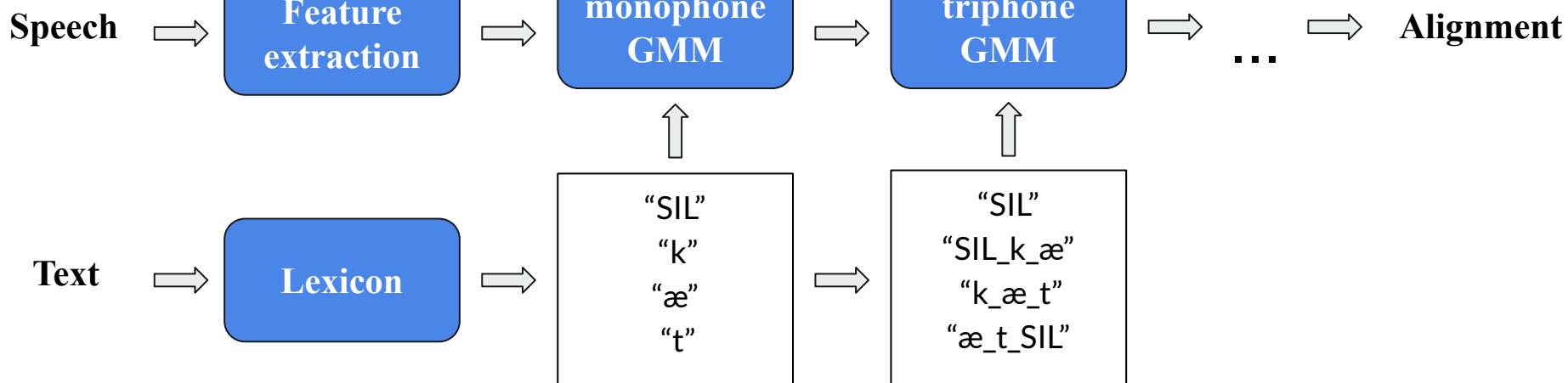
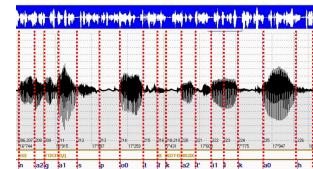
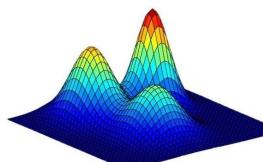
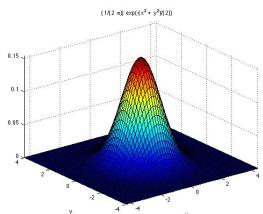
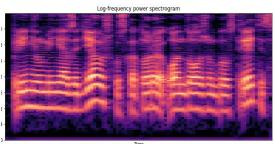


[ю]	, с вероятностью 0,09
[у]	, с вероятностью 0,6
[о]	, с вероятностью 0,01
...	

Phone alignment

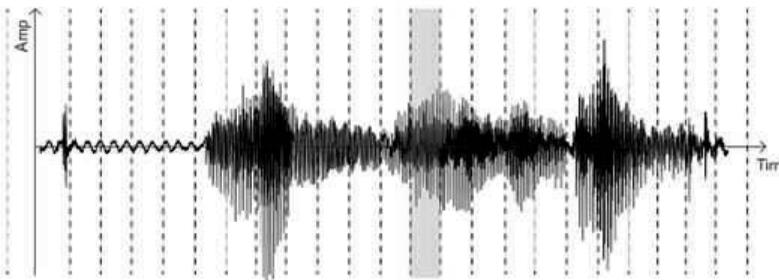


Force alignment

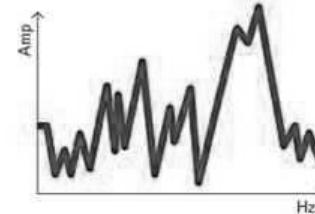


Acoustic model

1 Фреймы



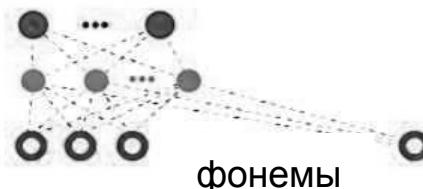
2 Спектр фрейма



3 MFCC

(9, 55.54, 8.113, 3.5553, -1.583, , 2.4, 6.105)

4 Нейронная сеть



5 Распределение вероятностей по сенонам

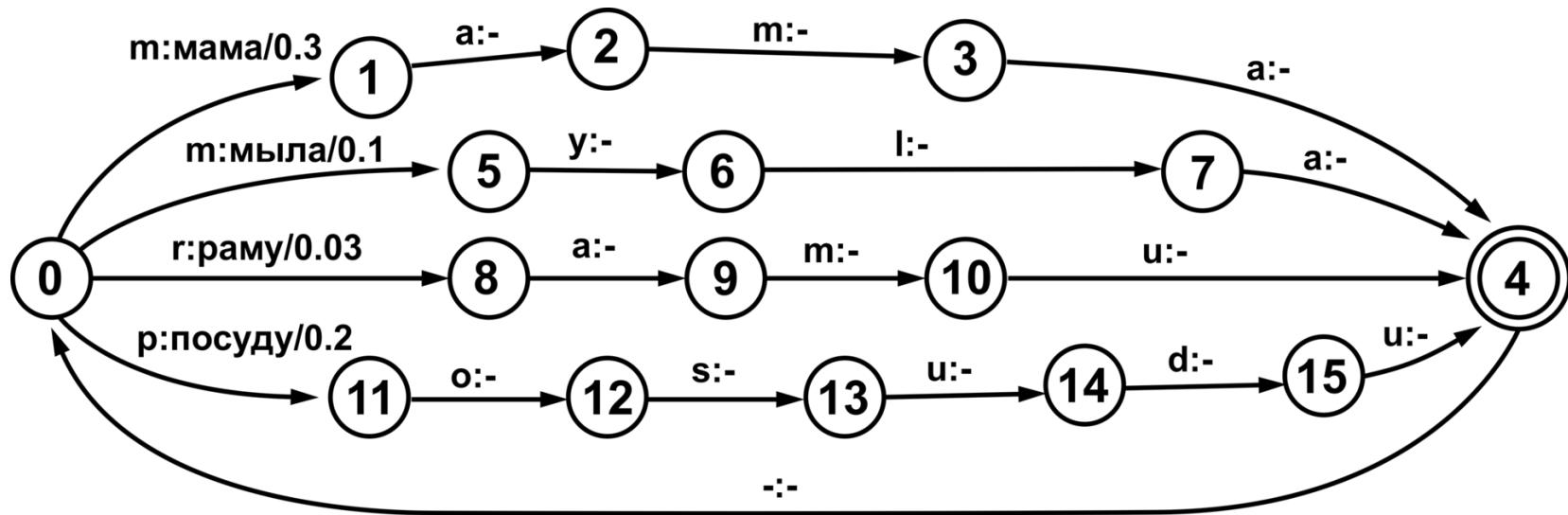


[ю]	, с вероятностью 0,09
[у]	, с вероятностью 0,6
[о]	, с вероятностью 0,01
...	

Что делать с выходами DNN?



Weighted Finite State Transducer



Мама мыла раму



контекст предсказание

$$w = (w_1, w_2, \dots, w_m)$$

$$\mathsf{P}(w) = \prod_{i=1}^m \mathsf{P}(w_i | w_{i-n+1} \dots w_{i-1})$$

Ngram arpa format

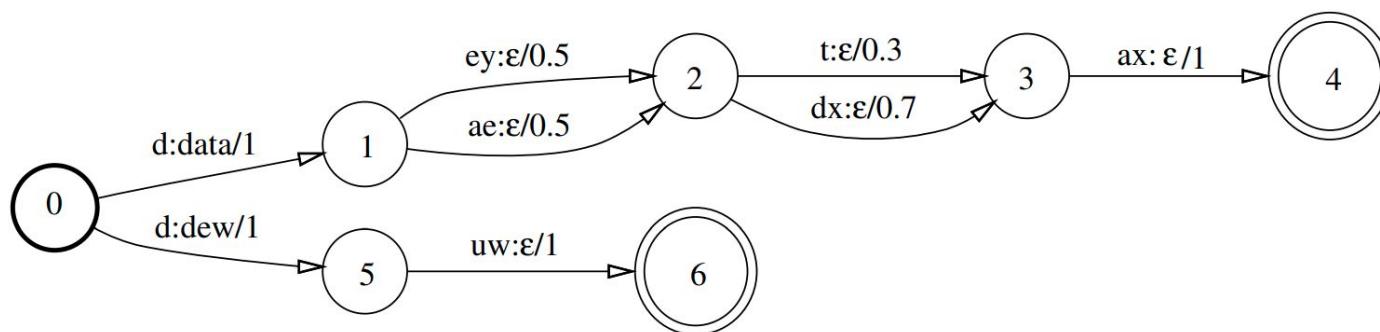
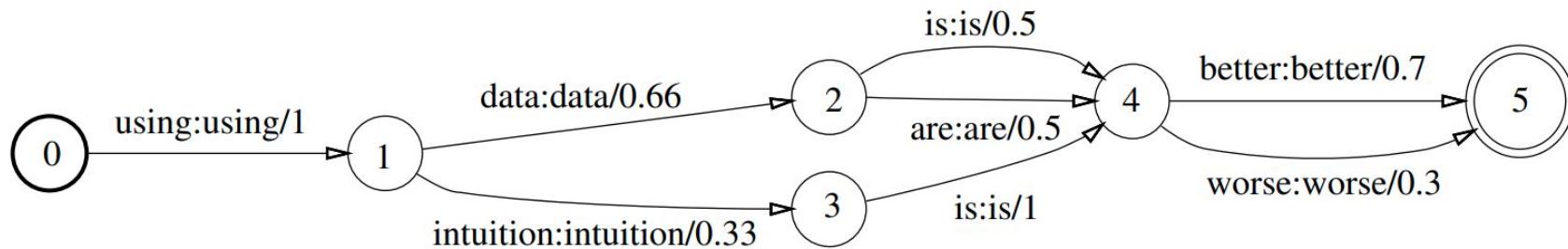
```
\data\
ngram 1=7
ngram 2=7

\1-grams:
-1.0000 <unk>      -0.2553
-98.9366 <s> -0.3064
-1.0000 </s>  0.0000
-0.6990 wood       -0.2553
-0.6990 cindy      -0.2553
-0.6990 pittsburgh -0.2553
-0.6990 jean        -0.1973

\2-grams:
-0.2553 <unk> wood
-0.2553 <s> <unk>
-0.2553 wood pittsburgh
-0.2553 cindy jean
-0.2553 pittsburgh cindy
-0.5563 jean </s>
-0.5563 jean wood

\end\
```

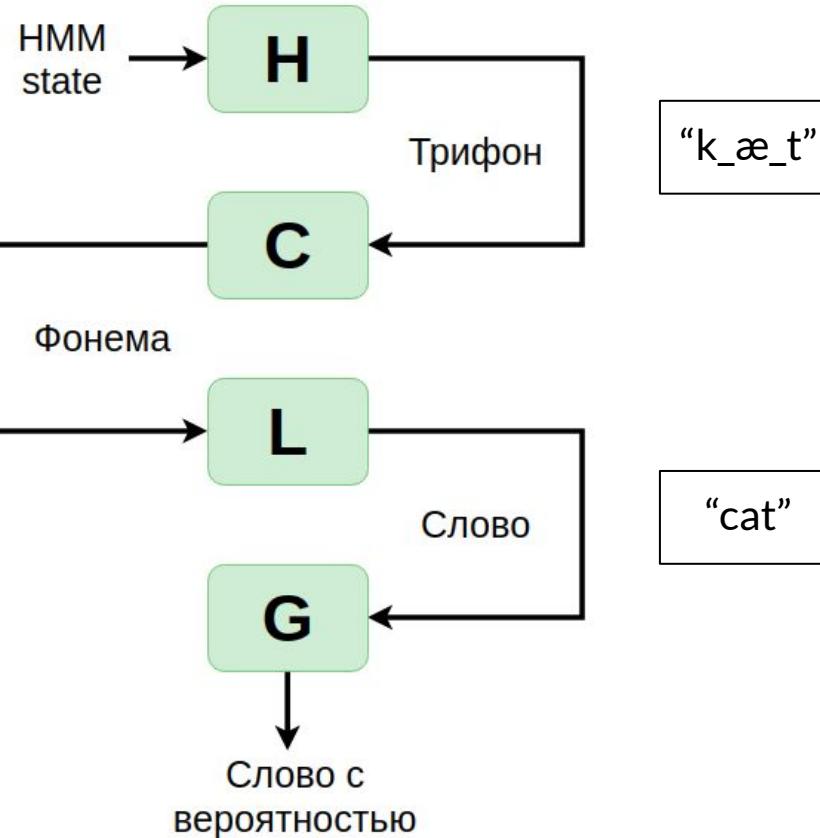
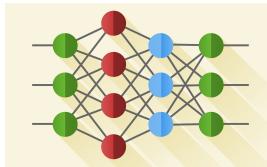
Граф декодирования



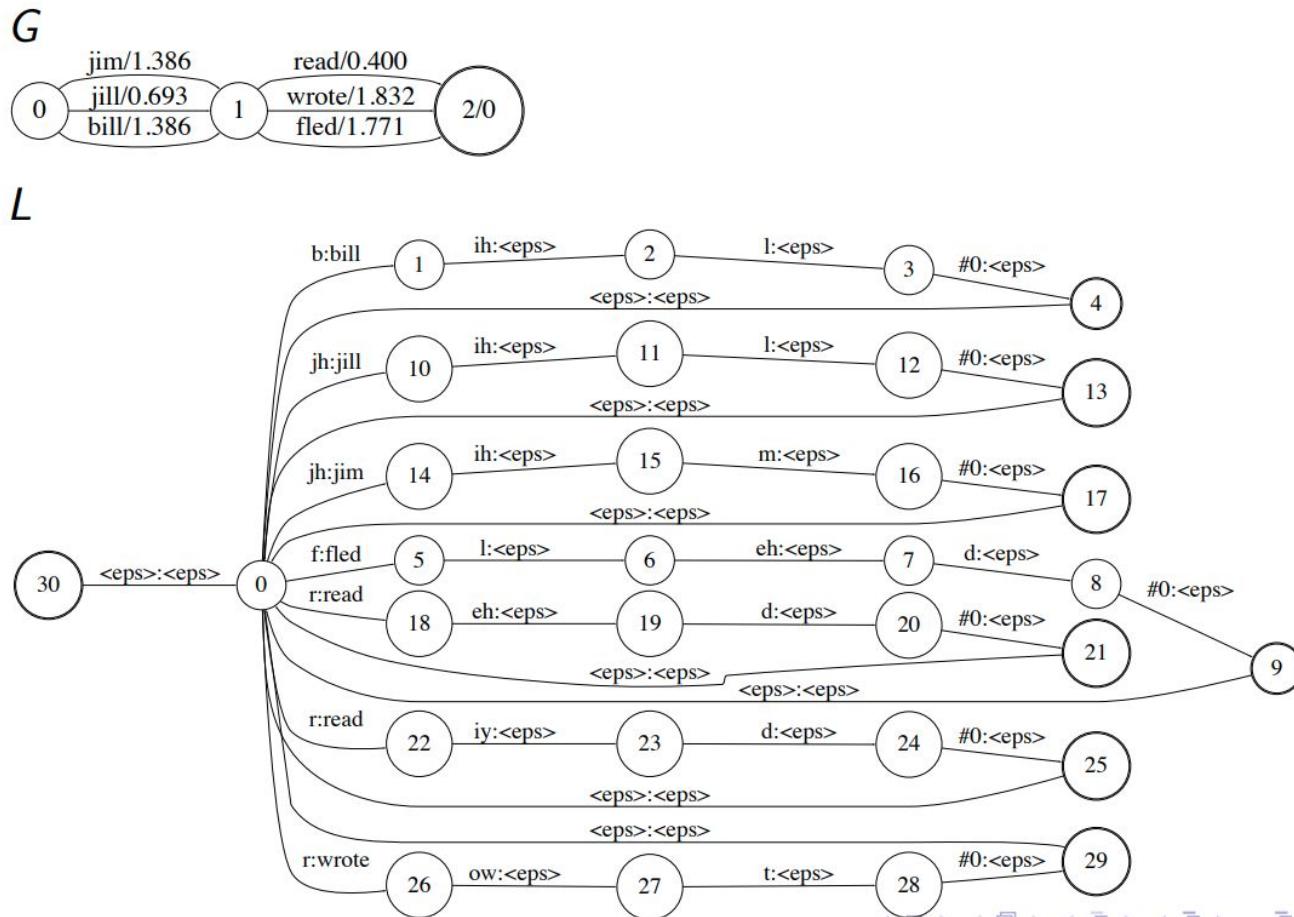
Стандартный граф декодирования в kaldi представляет собой HCLG граф, который является композицией четырех компонент: **H** \circ **C** \circ **L** \circ **G**:

	transducer	input sequence	output sequence
G	word-level grammar	words	words
L	pronunciation lexicon	phones	words
C	context-dependency	CD-phones	phones
H	HMM transition-ids	transition-ids	CD-phones

Граф декодирования



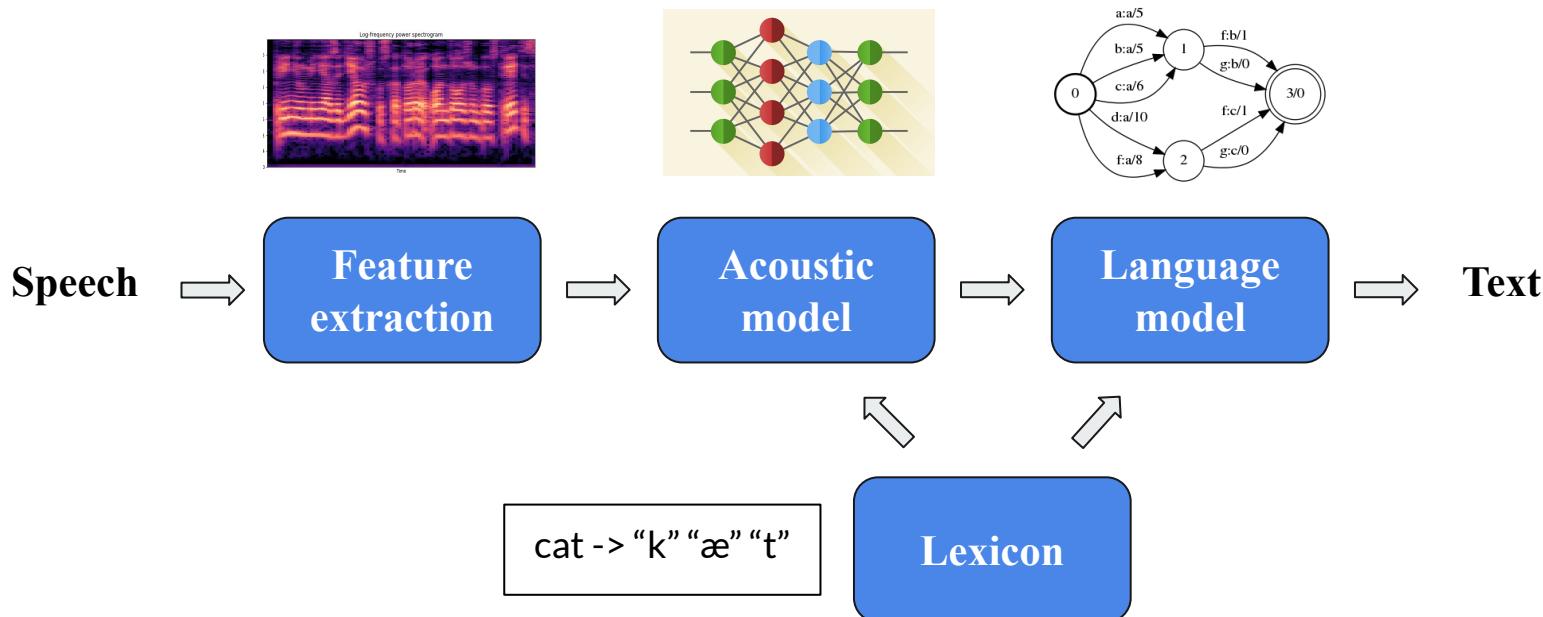
Граф декодирования



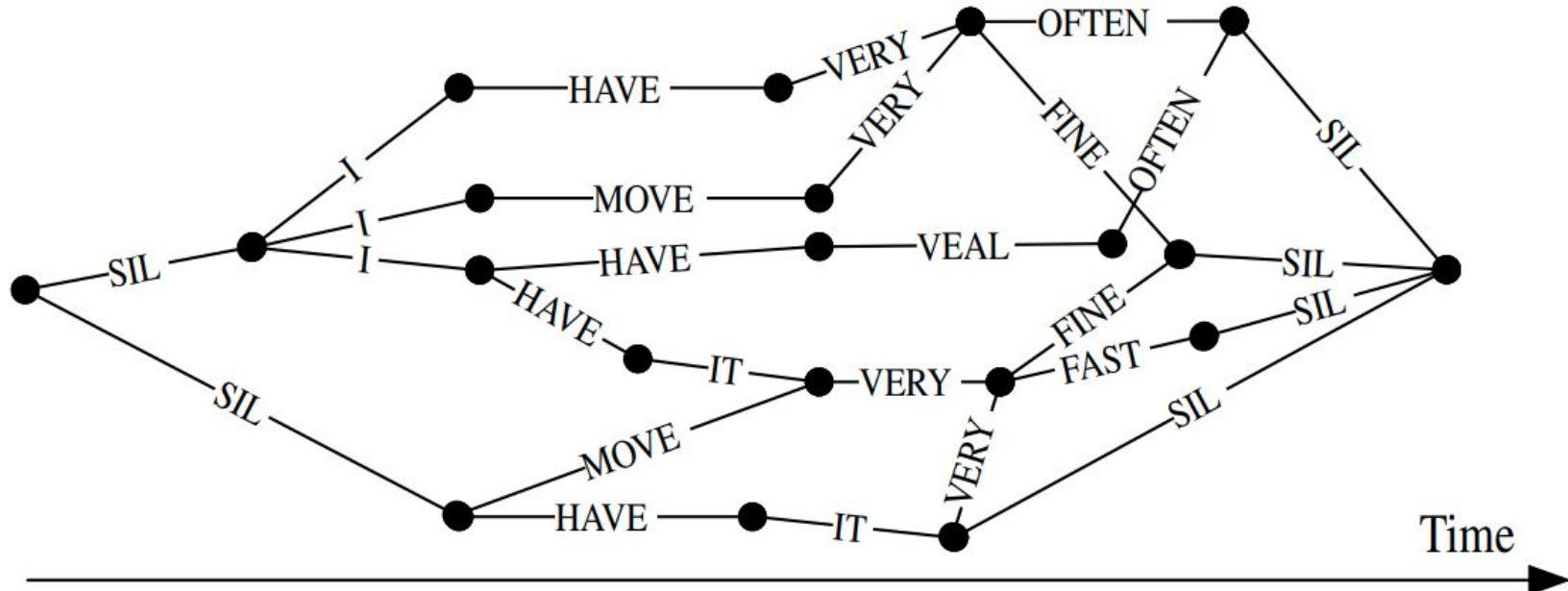
Граф декодирования



Классическая ASR система

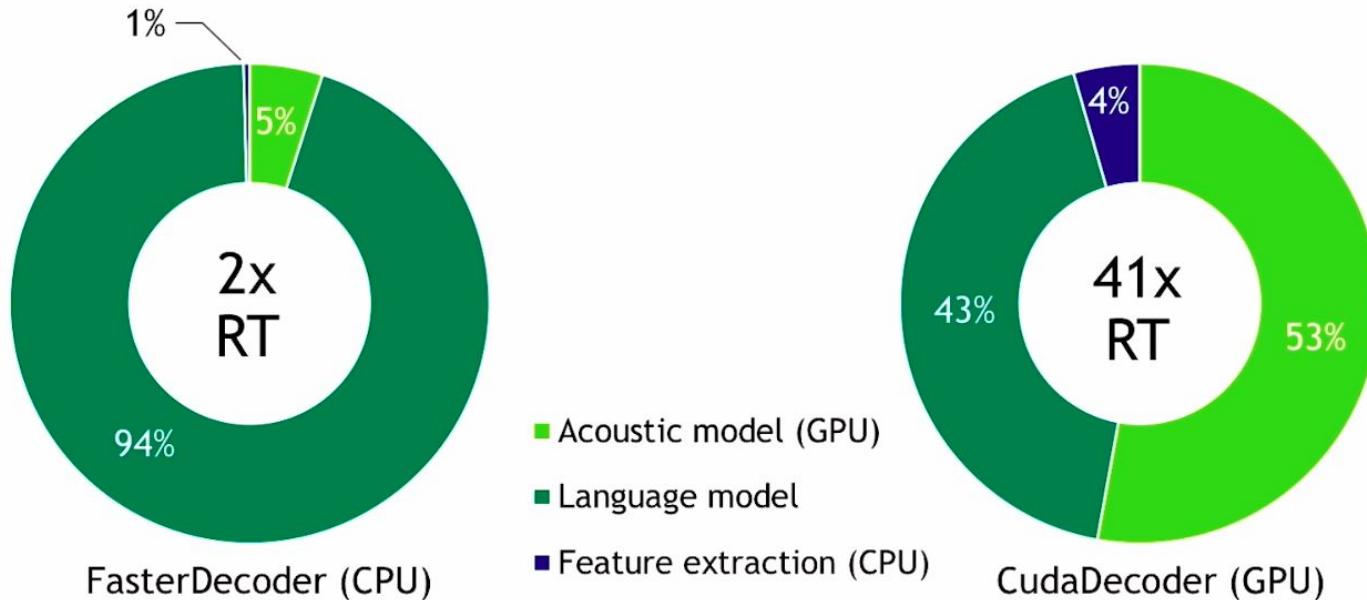


Word lattice



TIMING BREAKDOWN

ASpIRE, Chain Model, TDNN

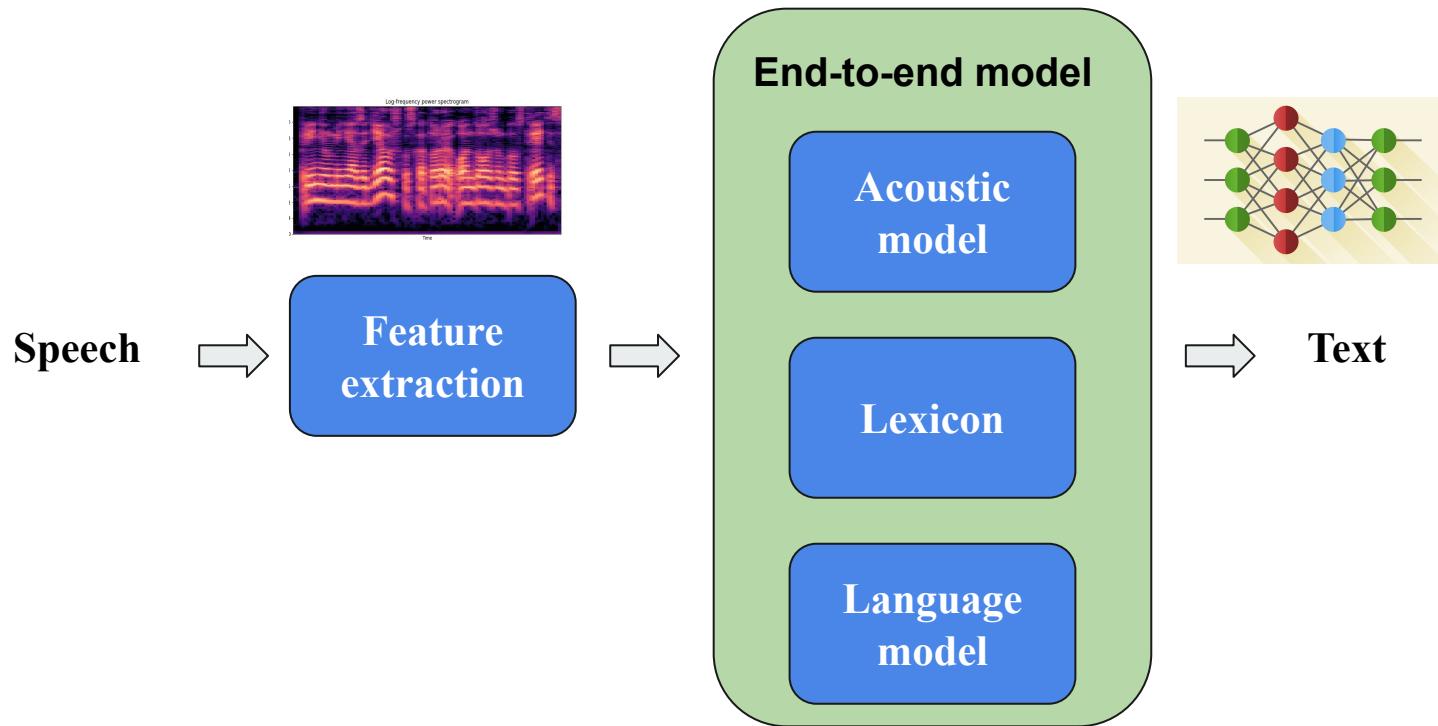


- Разделение на отдельные модули.
- Покадровая разметка на таргеты.
- Потребность в лексиконе.
- Ресурсоемкость декодирования по большому графу.
- Дополнительная конвертация моделей для использования в проде.
- Высокий порог входления в тему.

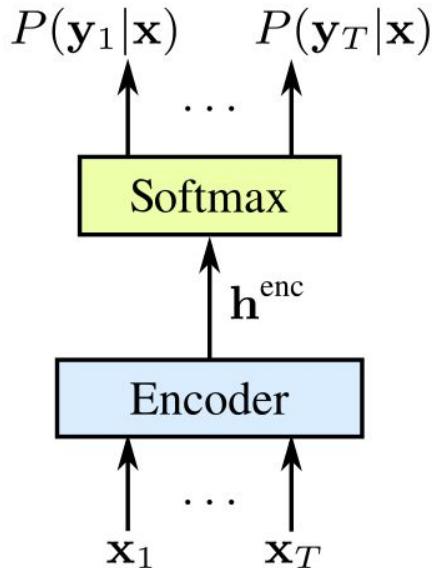
***** Все это значительно усложняет процесс построения ASR системы.**



End-to-end ASR system

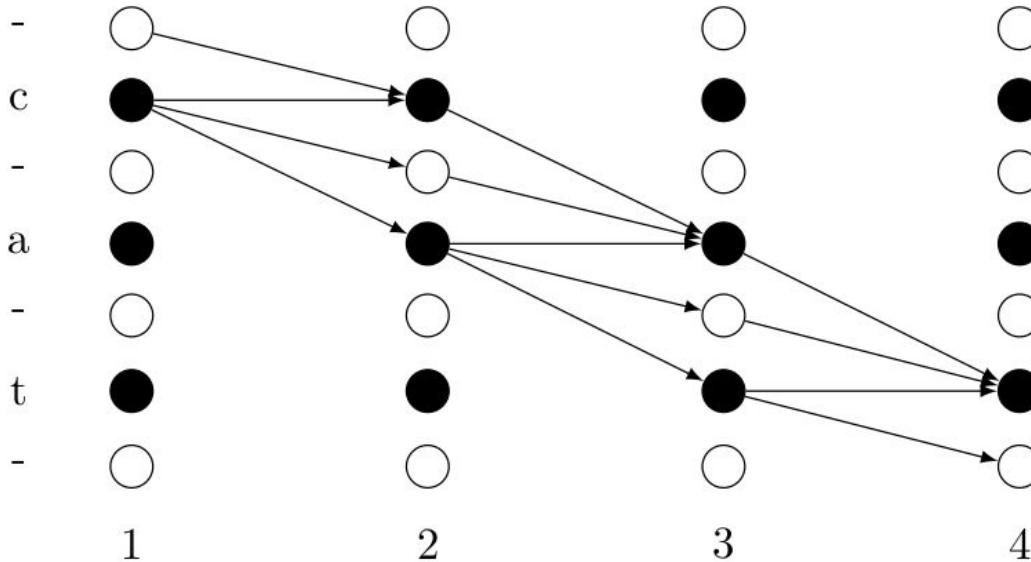


Connectionist Temporal Classification



(a.) CTC

Connectionist Temporal Classification

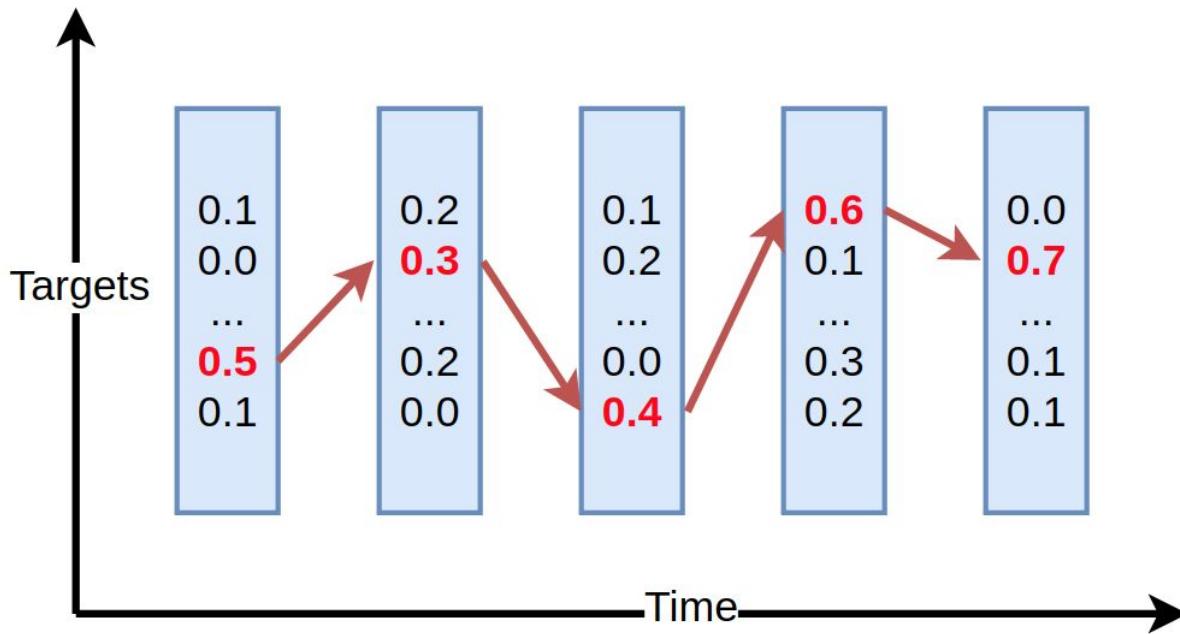


$\underbrace{(-, c, a, t), (c, -, a, t), (c, c, a, t), (c, a, -, t), (c, a, a, t), (c, a, t, -), (c, a, t, t)}_{'cat'}$

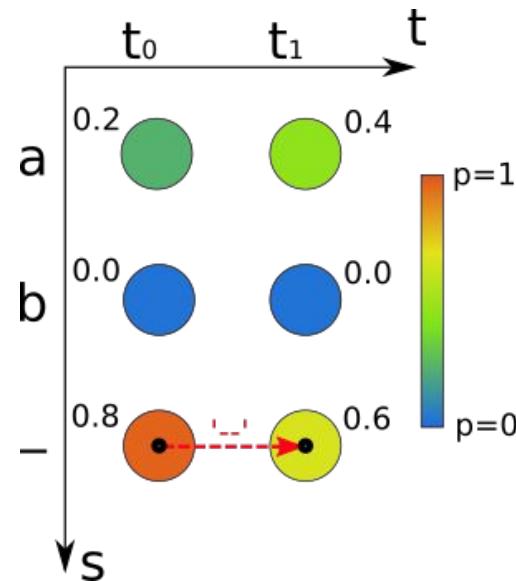
$$p(\mathbf{l}|\mathbf{x}) = \sum_{\pi \in \mathcal{B}^{-1}(\mathbf{l})} p(\pi|\mathbf{x}).$$

https://www.cs.toronto.edu/~graves/icml_2006.pdf

Greedy search decoding

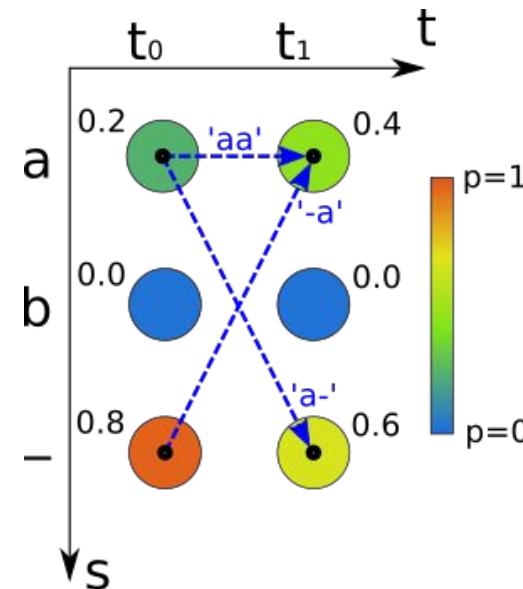


Beam search decoding



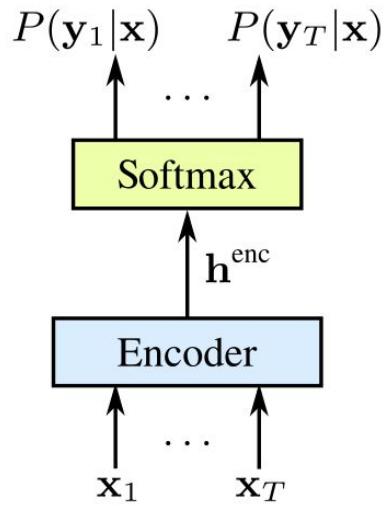
Greedy search

$$0.8 * 0.6 = 0.48$$

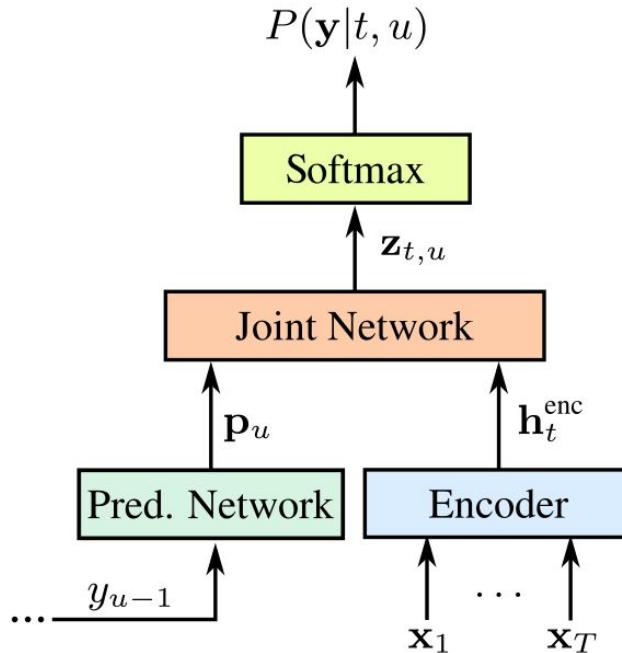


Beam search

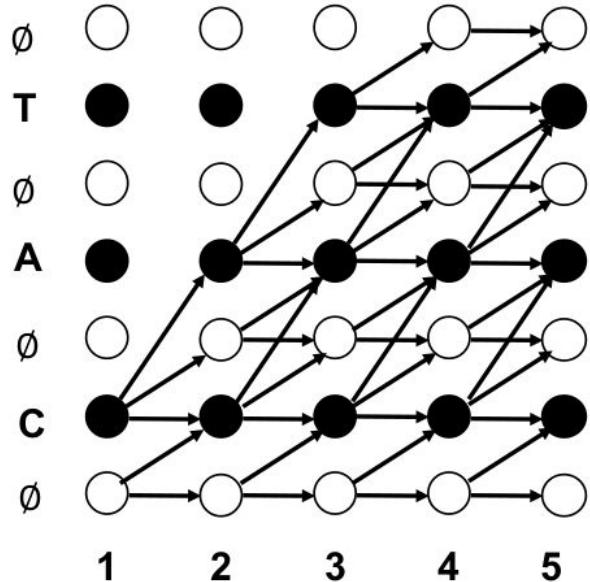
$$0.2 * 0.4 + 0.2 * 0.6 + 0.8 * 0.4 = 0.52$$



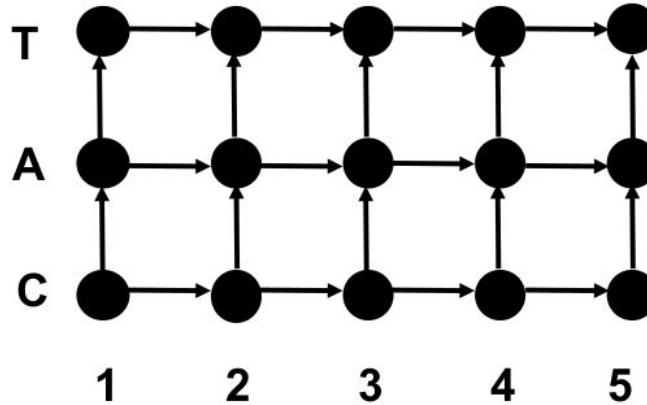
(a.) CTC



(b.) RNN-Transducer

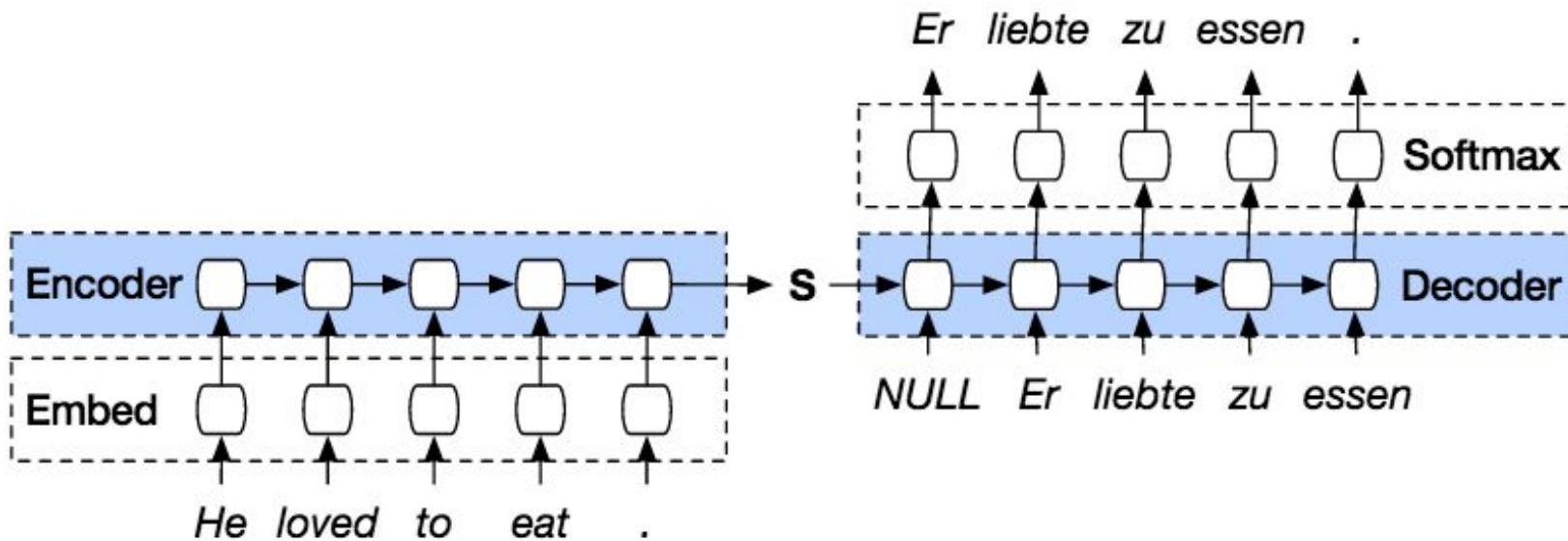


(a) CTC



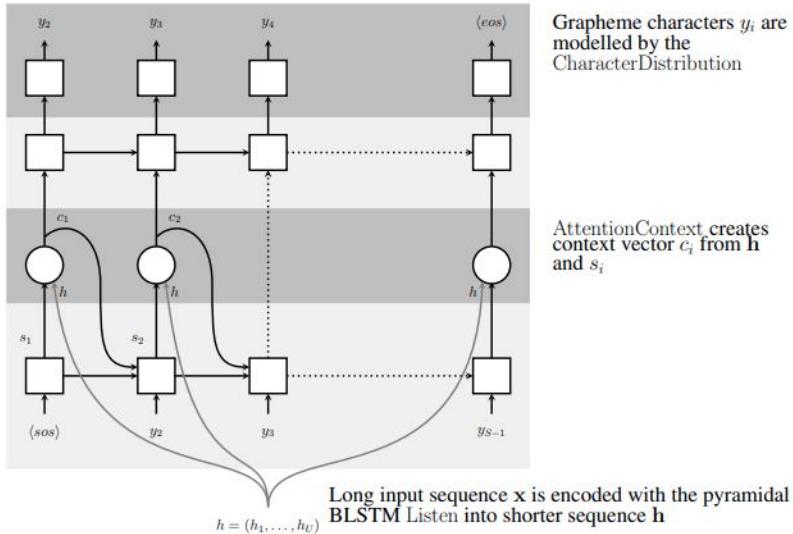
(b) RNN-Transducer

Seq2seq approach

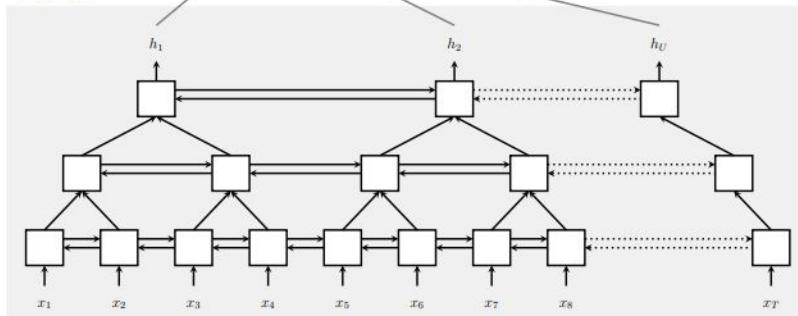


Listen attend and spell

Speller



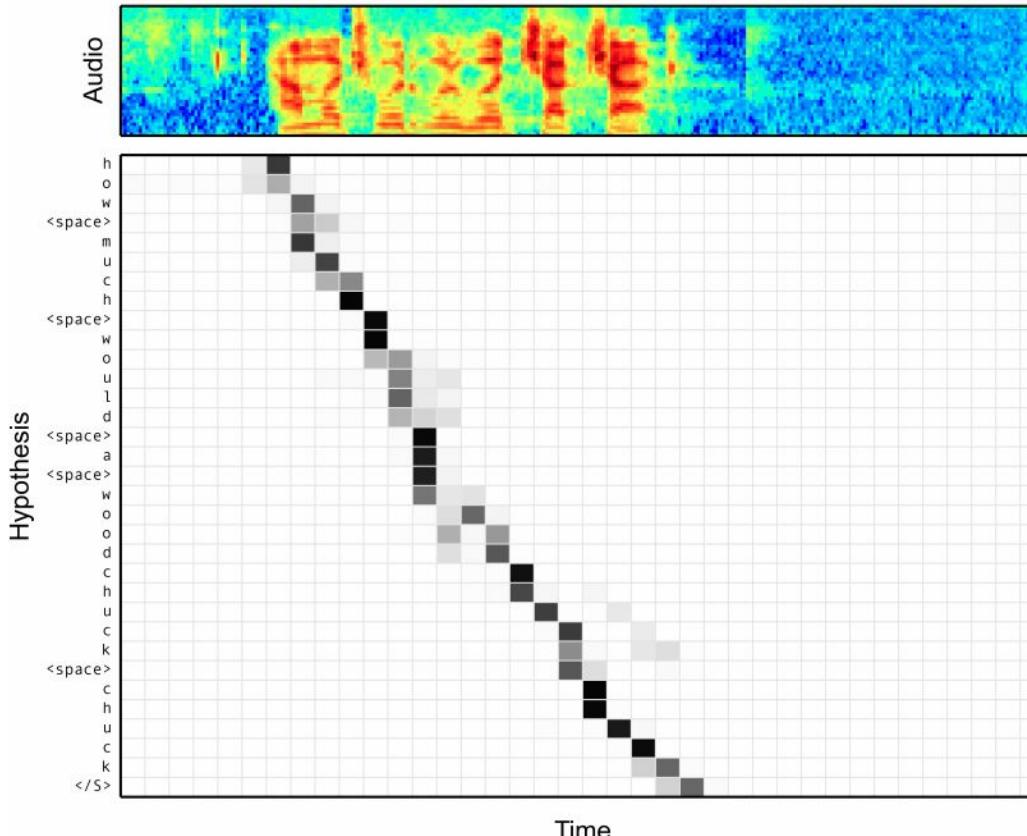
Listener



<https://arxiv.org/abs/1508.01211>

Listen attend and spell

Alignment between the Characters and Audio



Chunk-wise self-attention

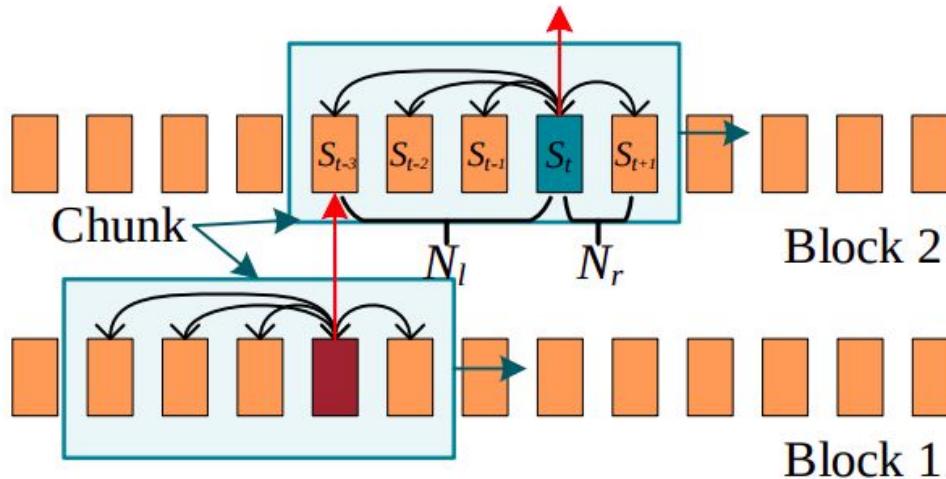


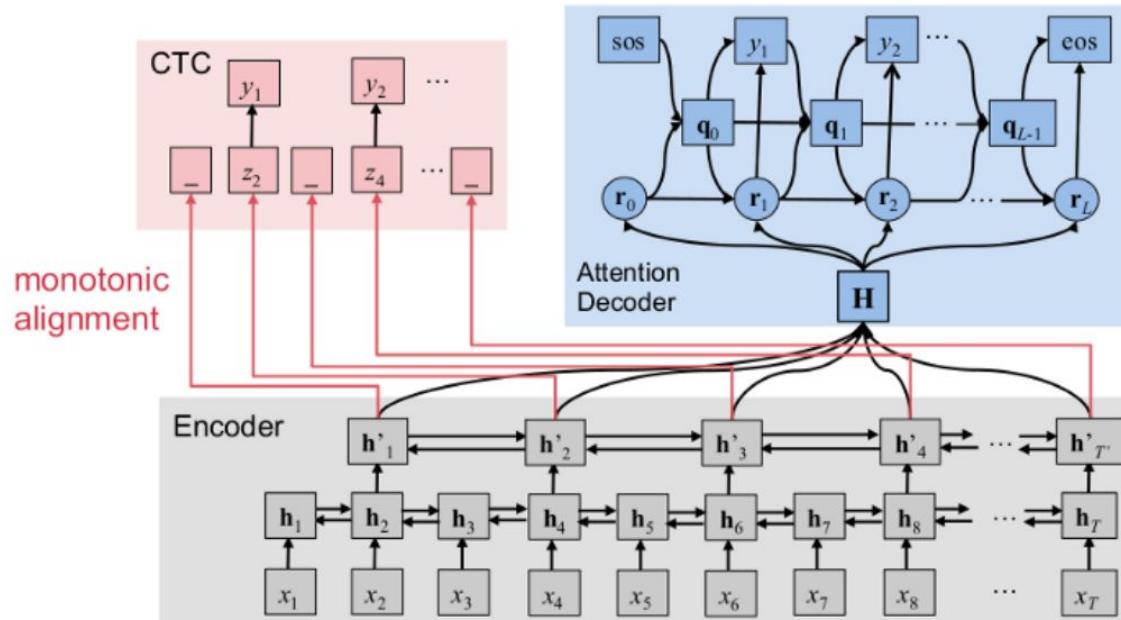
Figure 2: *Chunk-Flow Mechanism*. A blue box represents a chunk sliding along the time axis. Red arrows represent information flow between two layers and black arrows represent the calculation of self-attention weights.

Table 1. Model characteristics comparison.

Model	Delay	Computation Complexity	Language Model Ability	Training Difficulty	Recognition Accuracy
CTC-based	● ○ ○ ○ ○	● ○ ○ ○ ○	✗	● ○ ○ ○ ○	● ○ ○ ○ ○
RNN-Transducer	● ● ● ○ ○	● ● ● ● ●	✓	● ● ● ● ●	● ● ● ● ○
Attention-based	● ● ● ● ●	● ● ● ○ ○	✓	● ● ● ○ ○	● ● ● ● ●

<https://www.mdpi.com/2073-8994/11/8/1018>

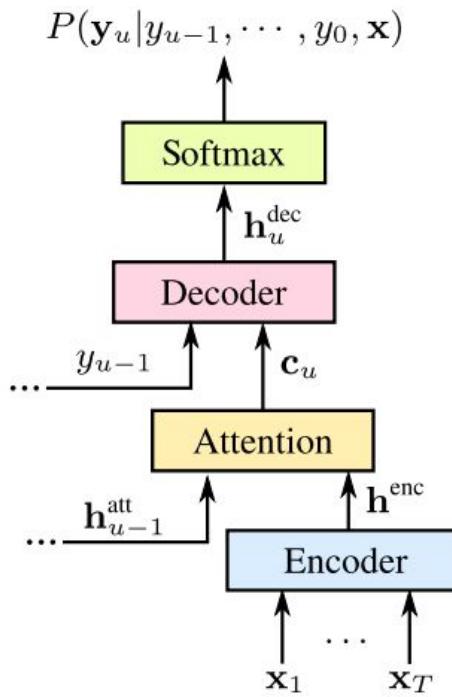
$$\text{Multitask learning: } \mathcal{L}_{\text{MTL}} = \lambda \mathcal{L}_{\text{CTC}} + (1 - \lambda) \mathcal{L}_{\text{Attention}}$$



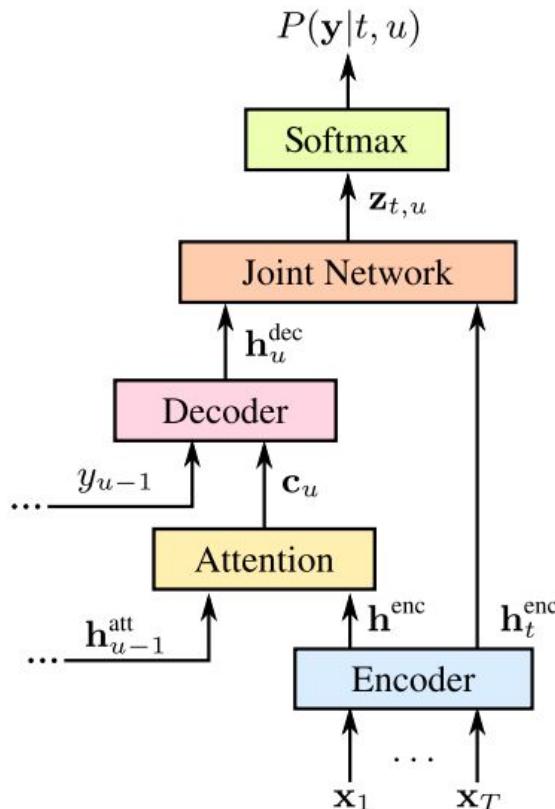
CTC guides attention alignment to be monotonic

Joint CTC/Attention architecture

Популярные end-to-end архитектуры

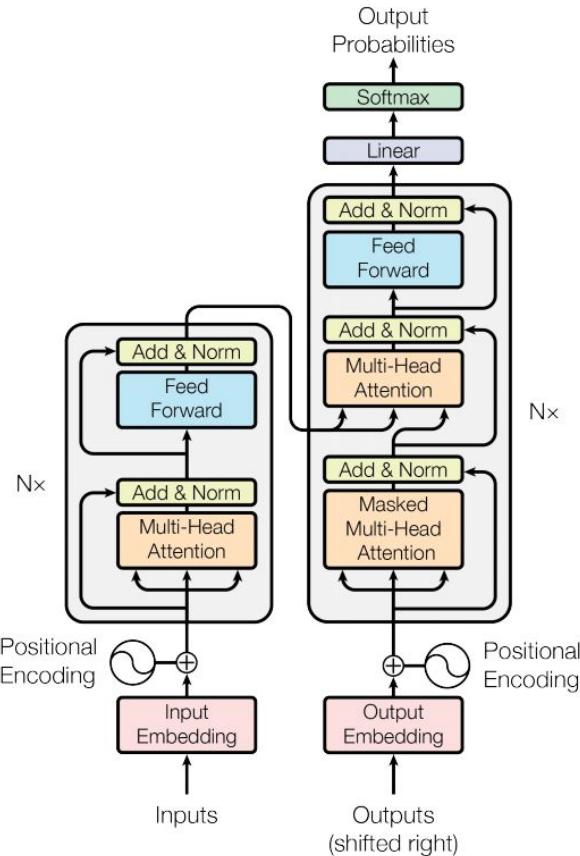


(c.) Attention-based Model



(d.) RNN-Transducer with Attention

Attention is all you need



<https://arxiv.org/abs/1706.03762>

Figure 1: The Transformer - model architecture.

- Потребность в большом количестве аудио данных (десятки тысяч часов).
- Вычислительные ресурсы.
- Низкая эффективность использования внешних языковых моделей.
- Проблемы с персонализацией (добавление новых слов).

RWTH ASR Systems for LibriSpeech: Hybrid vs Attention - w/o Data Augmentation

Christoph Lüscher¹, Eugen Beck^{1,2}, Kazuki Irie¹, Markus Kitzas¹, Wilfried Michel^{1,2}, Albert Zeyer^{1,2}, Ralf Schlüter¹, Hermann Ney^{1,2}

¹Human Language Technology and Pattern Recognition, Computer Science Department,
RWTH Aachen University, 52074 Aachen, Germany

²AppTek GmbH, 52062 Aachen, Germany

<https://arxiv.org/abs/1905.03072>

Hybrid vs end-to-end

Table 4: The WER results from our most interesting models and important results from other papers on LibriSpeech 960 h. CDp are (clustered) context-dependent phones. BPE are sub-word units. 4-gr LM is the official 4-gram word LM. GCNN are gated convolutional NN. RNN are recurrent NN.

paper	model	label unit		LM	WER [%]					
		AM	LM		dev		test			
					clean	other	clean	other		
Han et al. [3]	hybrid, seq. disc., single	CDp	word	RNN	3.0	8.8	3.6	8.7		
	hybrid, seq. disc., ensemble				2.6	7.6	3.2	7.6		
Zeghidour et al. [8]	end-to-end GCNN	chars	words	GCNN	3.2	10.1	3.4	11.2		
Irie et al. [9]	end-to-end attention	Word Piece Model	LSTM	LSTM	3.3	10.3	3.6	10.3		
Zeyer et al. [5]					3.5	11.5	3.8	12.8		
this work			BPE	None	4.3	12.9	4.4	13.5		
				LSTM	2.9	8.9	3.2	9.9		
				Transformer	2.6	8.4	2.8	9.3		
hybrid	CDp	word	4-gr	4.0	9.6	4.4	10.0			
hybrid, seq. disc.				3.4	8.3	3.8	8.8			
+ LSTM			2.2	5.1	2.6	5.5				
Transformer resc.			1.9	4.5	2.3	5.0				
Park et. al. [10]	end-to-end attention/SpecAugment		Word Piece Model		LSTM	-	-	2.5	5.8	

Speech Recognition on LibriSpeech test-clean



Exploration of End-to-End ASR for OpenSTT – Russian Open Speech-to-Text Dataset

Andrei Andrusenko^{1*}, Aleksandr Laptev^{1*}, and Ivan Medennikov^{1,2}

¹ ITMO University, St. Petersburg, Russia

² STC-innovations Ltd, St. Petersburg, Russia

{andrusenko, laptev, medennikov}@speechpro.com

Model	WER, %		
	calls	YouTube	books
TDNN-F LF-MMI	33.5	20.9	18.6
CTC-Attention	38.9	22.4	18.9
RNN-transducer	39.3	20.3	21.0
Transformer	34.8	19.1	16.8
Transformer [*]	32.6	20.8	18.1
Separable convolution & CTC [**]	37.0	26.0	23.0

Towards a Competitive End-to-End Speech Recognition for CHiME-6 Dinner Party Transcription

Andrei Andrusenko^{1,*}, Aleksandr Laptev^{1,*}, Ivan Medennikov^{1,2}

¹ITMO University, St. Petersburg, Russia

²STC-innovations Ltd, St. Petersburg, Russia

{andrusenko, laptev, medennikov}@speechpro.com

Model	WER(%)
Joint CTC/Attention E2E [15]	82.1
CNN-based Multichannel E2E [16]	80.7
CHiME-6 TDNN-F baseline [30]	51.7
RNN-T + <i>dev_gss12</i>	55.0
RNN-T + <i>train_gss</i> + <i>dev_gss12</i>	52.6
RNN-T + <i>train_gss</i> + <i>dev_gss24</i>	49.0
Hybrid system (n-gram LM) [25]	36.8
Hybrid system (AWD-LSTM-LM) [25]	33.8

Google on-device speech recognition

Задача:

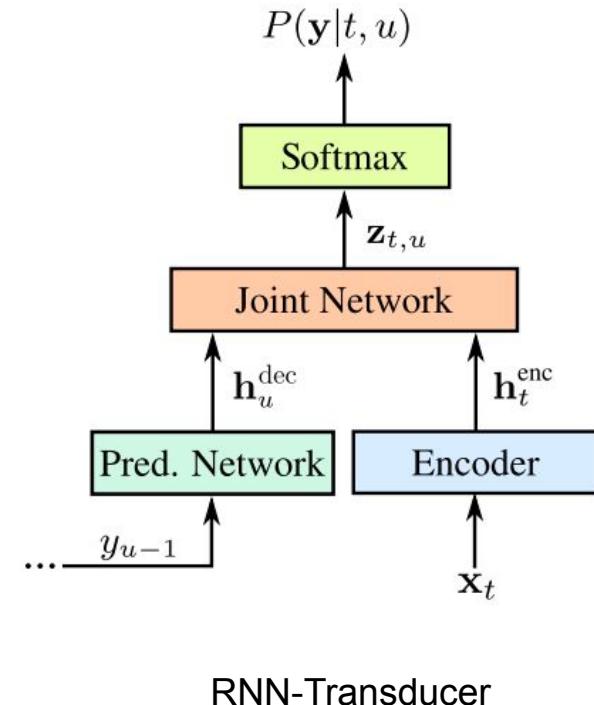
- Перенести распознавание речи на пользовательский девайс (Google Pixel).

Плюсы:

- Результаты распознавания доступны практически мгновенно.
- Разгрузка вычислительных центров.
- Приватность данных.

Минусы:

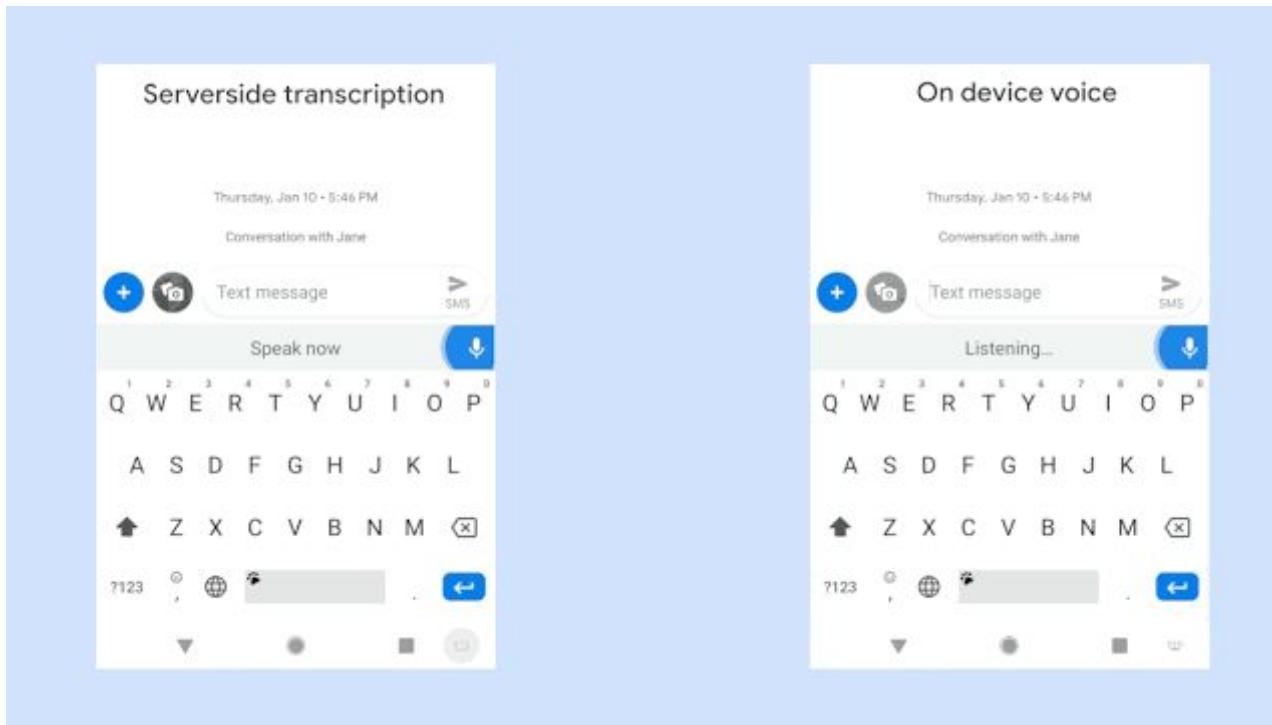
- Жесткие ограничения по размеру модели



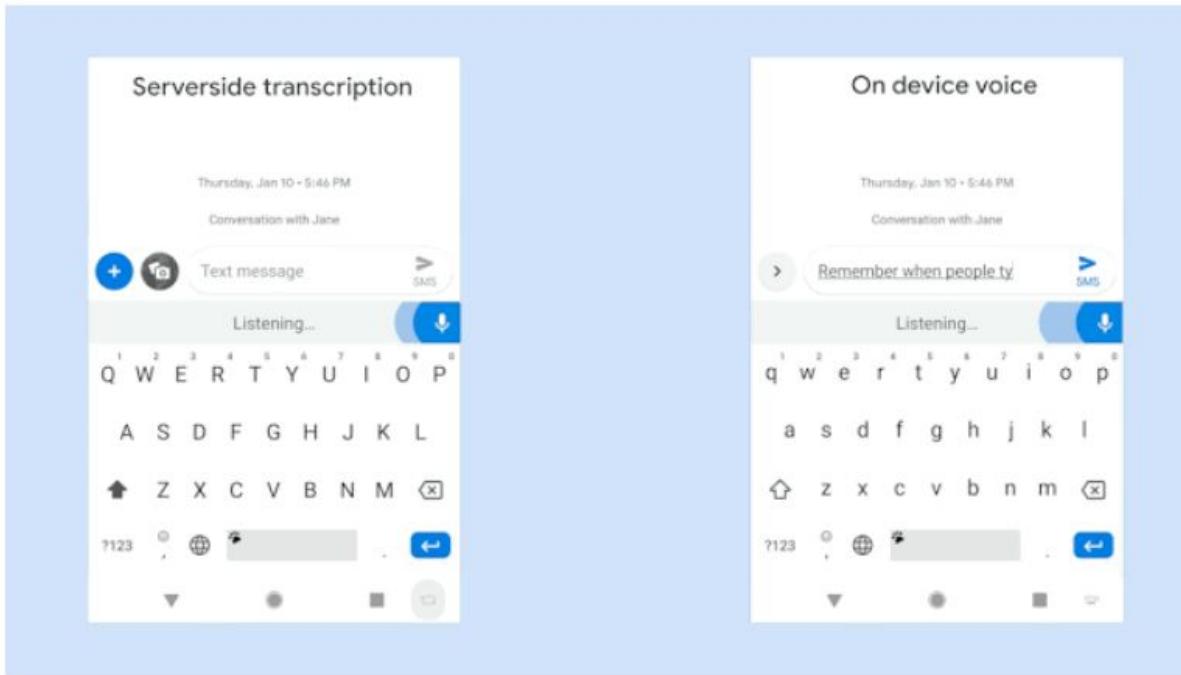
Additional features:

- Layer normalization to each LSTM layer.
- Word-pieces subword units.
- Efficient forward-backward algorithm (batched computation in tensorflow) which allows use TPUs for RNN-T (large batches, training faster than GPUs).
- Quantize parameters from 32-bit floating-point into 8-bit fixed-point.
- Contextual biasing (addition user's data: favorite songs, contacts etc) via shallow fusion with an external language model at inference time (at each step of the beam search).
- TTS data of numeric sequences for training (5 million utterances, 90% real and 10% synthetic data per batch).
- Train data ~ 27500 hours (This data set is created by room simulator, adding varying degrees of noise and reverberation such that the overall SNR is between 0dB and 30dB, with an average SNR of 12dB).

Google on-device speech recognition



Google on-device speech recognition

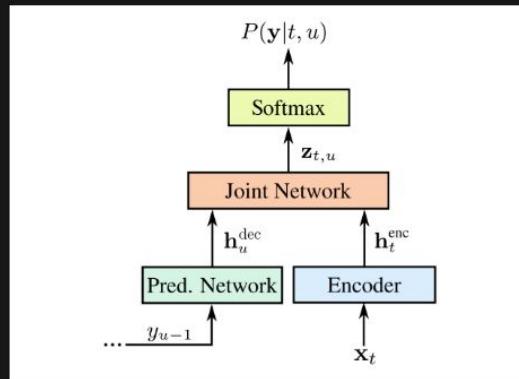


Reverse engineering

The workflow will be as follows:

1. Find the trained models (DONE)
2. Figure out how to import the model in TensorFlow (DONE)
3. Figure out how to connect the different inputs and outputs to each other (in progress)
4. (optional) export to lwttn
5. Write lightweight application for dictation
6. (stretch goal) if importing to TensorFlow Lite is successful, try to get it to work on those cool new RISC-V k210 boards, which could be had including 6 mic array for ~\$20!

Finding the trained models was done by reverse engineering the GBoard app using apktool. Further analysis of the app is necessary to find the right parameters to the models, but the initial blog post also provides some useful info:



Yandex



Сейчас:

- Jasper
- CTC-loss
- N-gramm LM
- QuartzNet
- CTC-loss
- N-gramm LM
- Transformer LM
- QuartzNet
- CTC-loss
- N-gramm LM + WFST

Развивают:

- Seq2seq
- Bert/GPT-3
- RNN-T

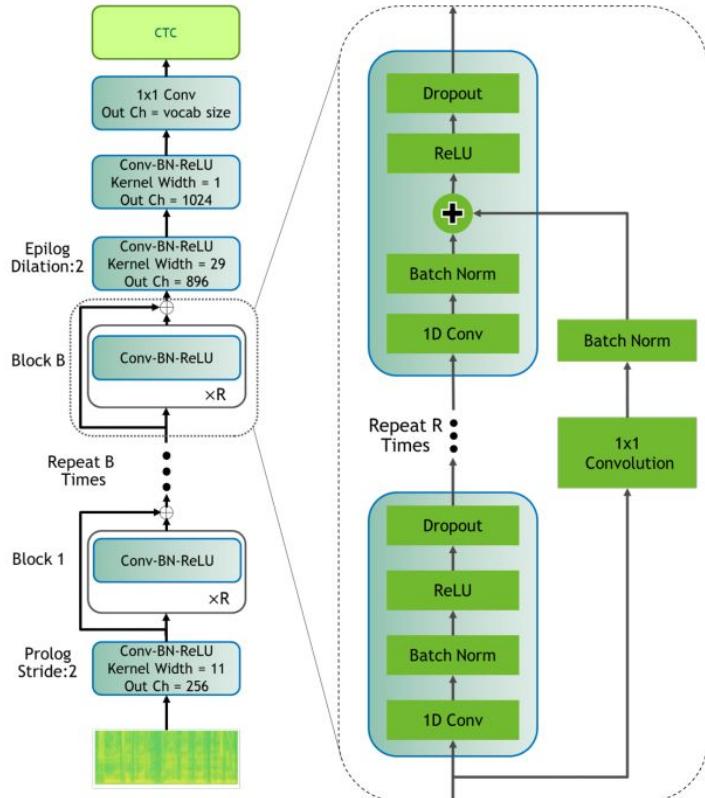


Figure 1: Jasper BxR model: B - number of blocks, R - number of sub-blocks.

<https://arxiv.org/abs/1904.03288>

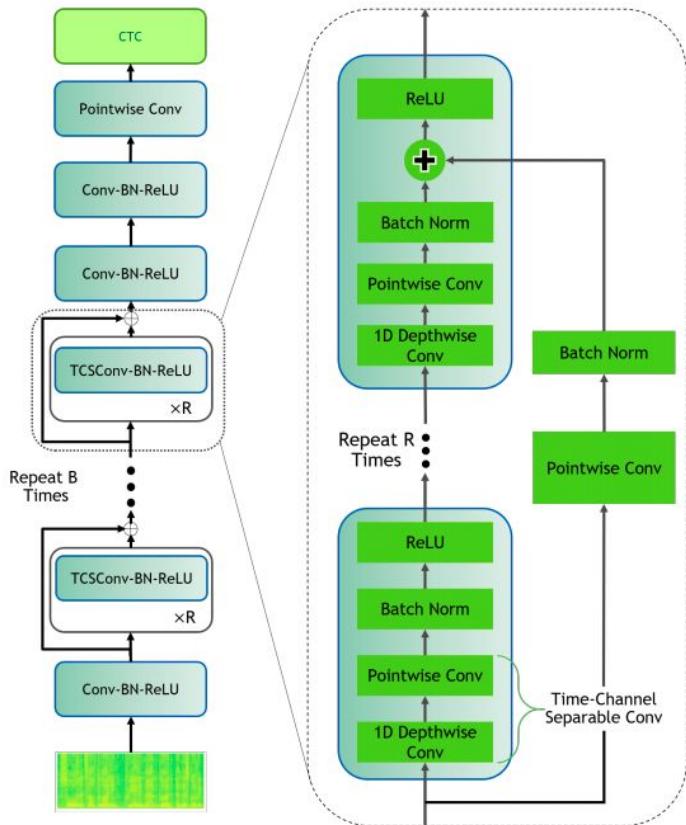
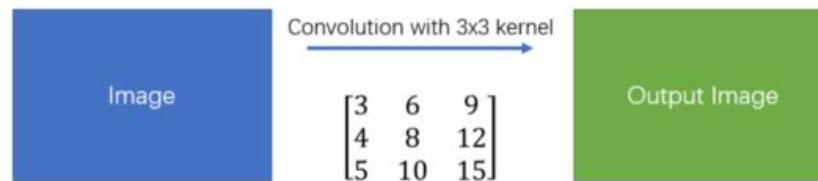


Fig. 1. QuartzNet BxR architecture

<https://arxiv.org/abs/1910.10261>

Simple Convolution



Spatial Separable Convolution



1,228,800 multiplications -> 53,952 multiplications



Ключевые особенности:

- Дискриминативное обучение гибридных моделей.
- Высокая скорость работы за счет C++ движка.
- Основной тулкит для гибридных систем.

Разработчики: Daniel Povey и другие.



<https://github.com/kaldi-asr/kaldi>

A man with dark hair and glasses, wearing a light blue t-shirt, stands in front of a brick wall. He is holding a large black protest sign with white, hand-painted text that reads "LET US GET BACK TO WORK". To his right, another person holds a white sign with green and purple text that partially reads "WELCOME ON AQUI".

LET US
GET BACK
TO WORK

Video of May 7-8 Attack on the Garland Sit-in & Occupation by Dr. Daniel Povey...

Imechapishwa na JHU Sit-In

Imetazamwa mara 8,411





Распознавание речи
724 members

Андрей
A что вы древний калди до сих пор ковыряете? 12:06

С калди я ещё 2 года назад плясал две недели, еле как собрал..
wer для моей задачи никакой (телефонные разговоры по
сотовой сети).. Может с предобработкой следовало бы
поплясать, но что-то он меня разочаровал тогда и на
качественных аудиозаписях.. Думал он давно уже мёртв, а нет,
до сих пор ковыряют) edited 12:35



Ключевые особенности:

- Обучение end-to-end ASR (e2e LF-MMI за счет PyChain).
- Есть фичи из Kaldi.
- Python и PyTorch.
- Высокая скорость работы за счет fairseq.

Разработчики: ...



<https://github.com/freewym/espresso>

Ключевые особенности:

- Обучение end-to-end ASR (CTC, seq2seq).
- Высокая скорость работы за счет C++ движка (fairseq).

Разработчики: Facebook Research.



<https://github.com/facebookresearch/wav2letter>

Ключевые особенности:

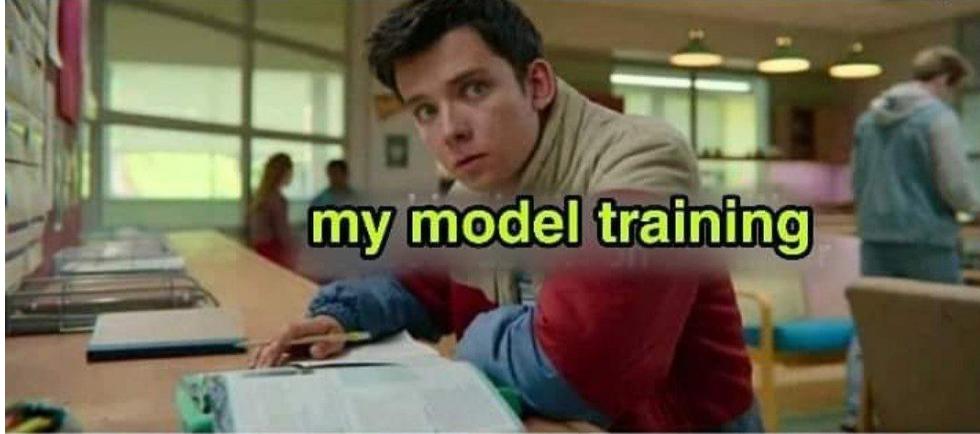
- Обучение end-to-end ASR (CTC, RNN-T, Seq2Seq).
- Большое количество готовых рецептов обучения на разных базах.
- Развитое сообщество.
- Удовлетворительная скорость работы за счет python кода.

Разработчики: Shinji Watanabe и другие.



<https://github.com/espnet/espnet>

Ожидание







INTERSPEECH 2019 tutorial

[espnet / interspeech2019-tutorial](https://github.com/espnet/interspeech2019-tutorial) Watch ▾ 11 Star 116 Fork 26

Code Issues 0 Pull requests 0 Actions Projects 0 Wiki Security Insights

INTERSPEECH 2019 Tutorial Materials

speech-recognition text-to-speech interspeech2019 tutorial

39 commits 2 branches 0 packages 0 releases 3 contributors

Branch: master New pull request Create new file Upload files Find file Clone or download ▾

kan-bayashi Update interspeech2019_tts.ipynb ✓ Latest commit 4ac07a3 on Sep 23, 2019

notebooks Update interspeech2019_tts.ipynb 5 months ago

tools cleaned up directory structure 6 months ago

.gitignore meetup slides 7 months ago

README.md Update README.md 6 months ago

README.md

Advanced methods for neural end-to-end speech processing – unification, integration, and implementation, INTERSPEECH2019 Tutorial (T6)

This repository provides the materials for INTERSPEECH 2019 Tutorial Advanced methods for neural end-to-end speech processing – unification, integration, and implementation.

<https://github.com/espnet/interspeech2019-tutorial>

Какие-то выводы



mail: andrusenkoau@gmail.com
telegram: @andrusenkoau

Andrusenko Andrei
Research scientist at STC

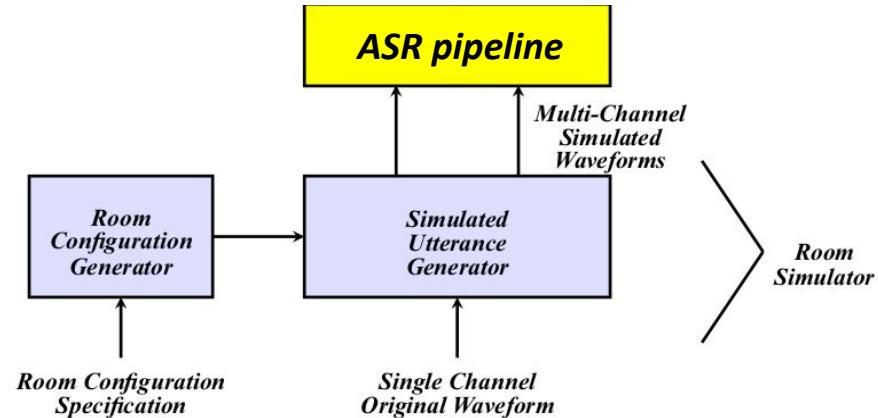
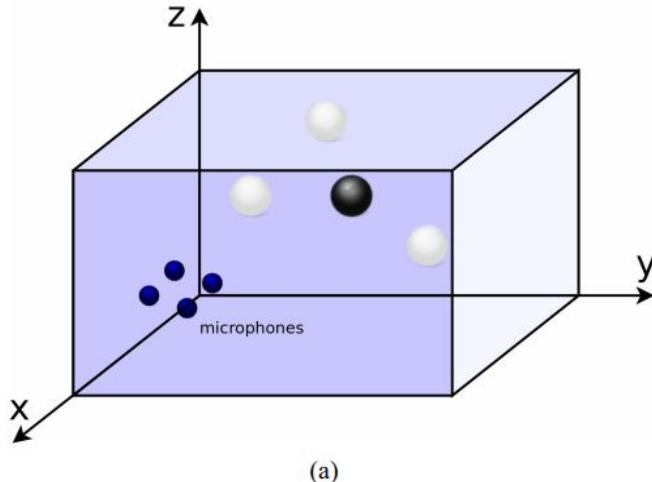
Задача:

- Увеличить количество данных.
- Сделать модель более стойкой к изменениям сигнала.

Решение:

- Speed and volume perturbation.
- Mean and variance normalization.
- SpecAugment.
- Room Impulse Response (RIR) data.
- Wav2vec.
- TTS data.

RIR data generation



A simulated room: There may be multiple microphones, a single target sound source, multiple noise sources in a cuboid-shape room with acoustically reflective walls.

