Yandex

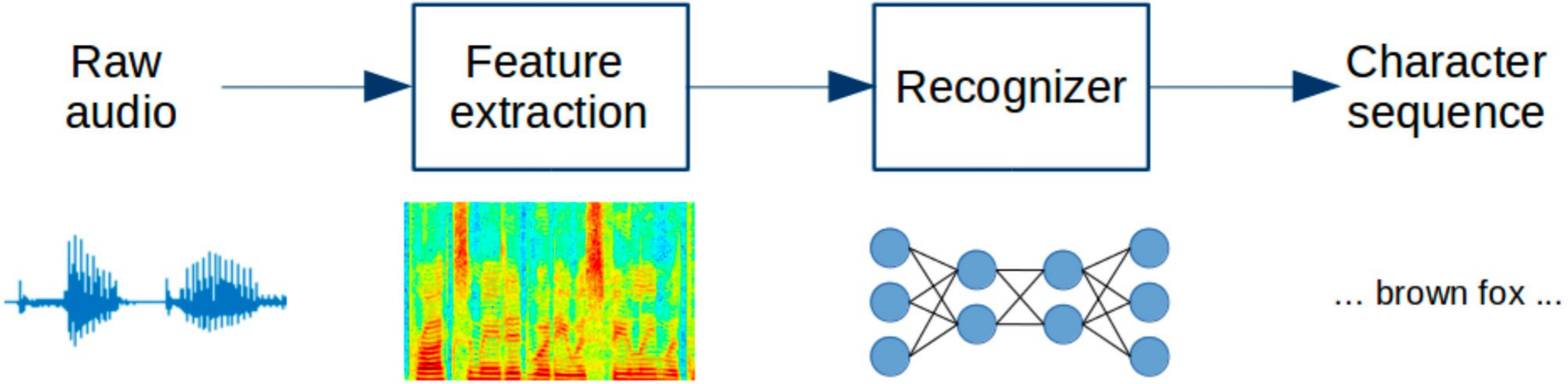
ШКОЛА АНАЛИЗА ДАННЫХ

Pretraining for ASR

Content

- **Recap**
- **Motivation:** why pretraining is useful for audio domain?
- **Data:** how **large** audio **unsupervised** and **supervised datasets** could be gathered?
- **Models and losses:** overlook how audio pretraining works
- **Evaluation:** how pretraining effectiveness could be measured?

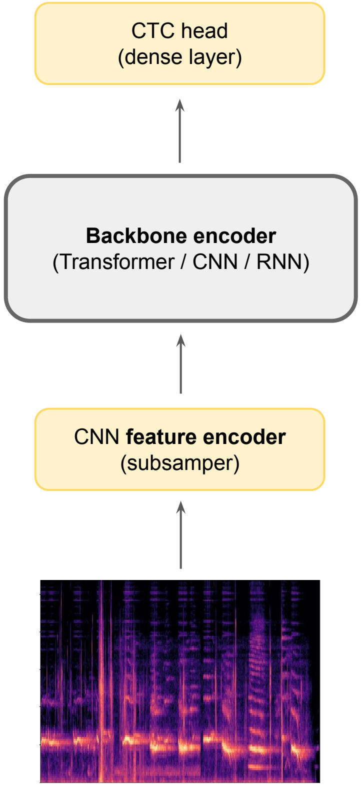
Recap: Speech recognition



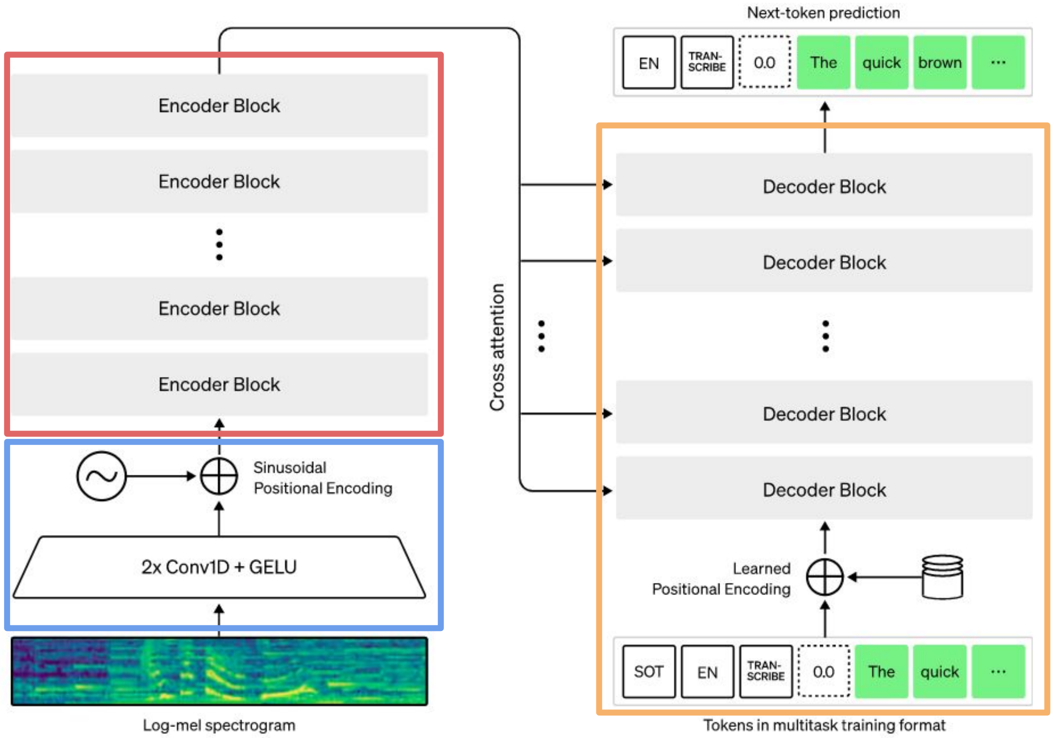
https://doi.org/10.3390/electronics9071157

Recap: CTC

Architecture



Recap: Seq2Seq



Components:

- CNN feature encoder (subsamper)
- Transformer encoder (could be RNN/CNN)
- Text transformer decoder (could be RNN/CNN)

https://arxiv.org/abs/2212.04356

Motivation: unlabeled data

Google DeepMind Gopher (280B params) pretraining Datasets

	Disk Size	Documents	Tokens	Sampling proportion
MassiveWeb	1.9 TB	604M	506B	48%
Books	2.1 TB	4M	560B	27%
C4	0.75 TB	361M	182B	10%
News	2.7 TB	1.1B	676B	10%
GitHub	3.1 TB	142M	422B	3%
Wikipedia	0.001 TB	6M	4B	2%

OpenAI GPT3 (175B params) pretraining datasets

Dataset	Quantity (tokens)	Weight in training mix	Epochs elapsed when training for 300B tokens
Common Crawl (filtered)	410 billion	60%	0.44
WebText2	19 billion	22%	2.9
Books1	12 billion	8%	1.9
Books2	55 billion	8%	0.43
Wikipedia	3 billion	3%	3.4

https://arxiv.org/abs/2005.14165
https://arxiv.org/abs/2112.11446

Motivation: unlabeled data

Dataset	Language	Total Duration (h)	Domain	Speech Type	Labeled	Label Type
Common Voice [1]	th	172.0	Open domain	Read	Yes	Manual
	id	28.0				
	vi	6.0				
FLEURS [10]	th	13.3	Wikipedia	Read	Yes	Manual
	id	12.6				
	vi	13.3				
VoxLingua107 [44]	th	61.0	YouTube	Spontaneous	No	-
	id	40.0				
	vi	64.0				
CMU Wilderness [4]	th	15.6	Religion	Read	Yes	Manual
	id	70.9				
	vi	9.2				
BABEL [13]	vi	87.1	Conversation	Spontaneous	Yes	Manual
VietMed [27]	vi	16.0	Medical	Spontaneous	Yes	Manual
Thai Dialect Corpus [41]	th	840.0	Open domain	Read	Yes	Manual
TITML-IDN [40]	id	14.5	News	Read	Yes	Manual
MEDISCO [36]	id	10.0	Medical	Read	Yes	Manual
YODAS manual [29]	th	497.1	YouTube	Spontaneous	Yes	Manual
	id	1420.1				
	vi	779.9				
YODAS automatic [29]	th	1.9	YouTube	Spontaneous	Yes	Pseudo
	id	8463.6				
	vi	9203.1				
GigaSpeech 2 raw	th	12901.8	YouTube	Spontaneous	Yes	Pseudo
	id	8112.9				
	vi	7324.0				
GigaSpeech 2 refined	th	10262.0	YouTube	Spontaneous	Yes	Pseudo
	id	5714.0				
	vi	6039.0				

https://arxiv.org/pdf/2406.11546

Motivation: audio foundation model

SUPERB

Recognition tasks
(Phoneme recognition, ASR)

Speaker tasks
(Speaker identification, speaker verification, speaker diarization)

Detection tasks
(Keyword spotting)

Paralinguistics tasks
(emotion classification)

Semantics tasks
(Speech translation, intent classification, slot filling)

Generation tasks
(Speech enhancement, speech separation)

https://superbbenchmark.org

Motivation: robust model

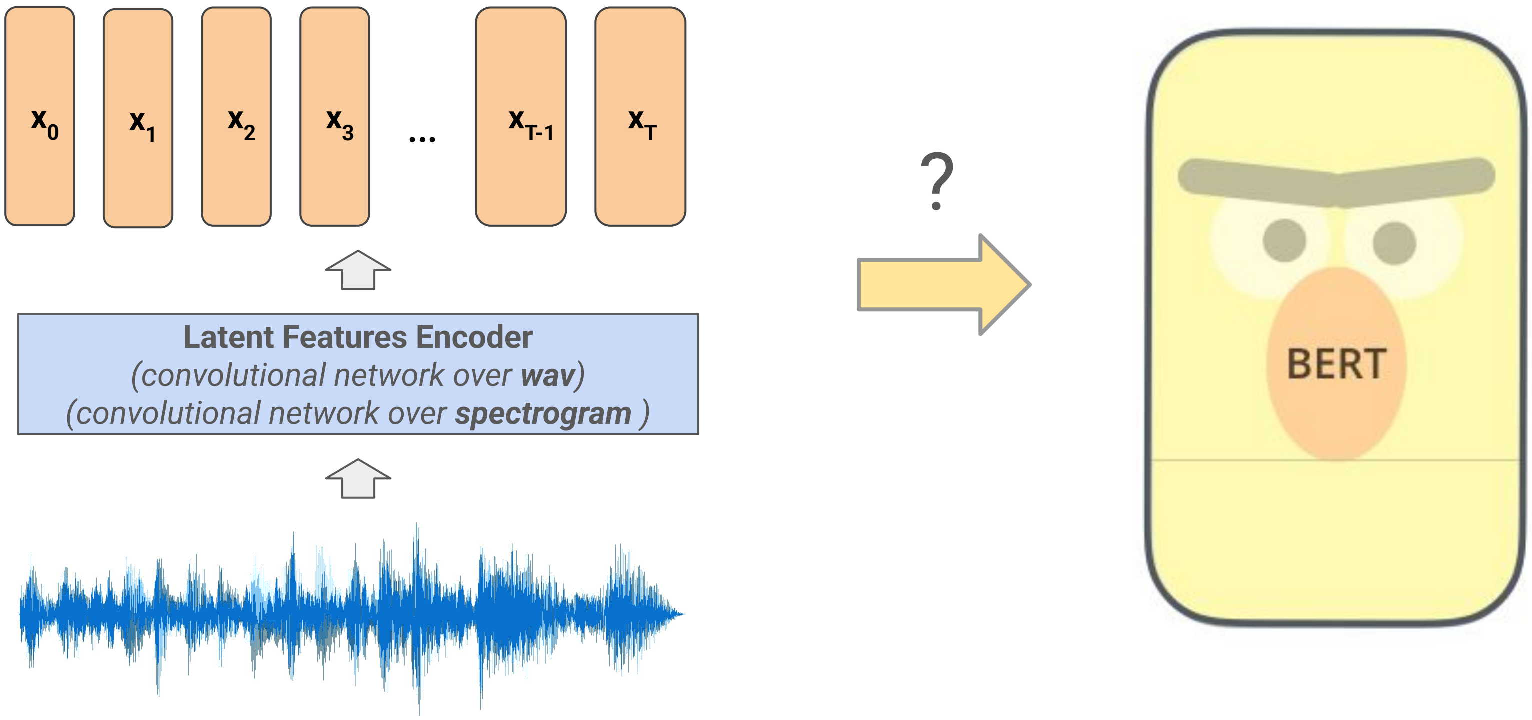
Robustness:

- overall – performance, average across both out and in domain datasets
- effective [Taori et al. (2020)] – measuring performance, compared to reference in domain dataset

Dataset	wav2vec 2.0 Large (no LM)	Whisper Large V2
LibriSpeech Clean	**2.7**	**2.7**
Artie	24.5	**6.2**
Common Voice	29.9	**9.0**
Fleurs En	14.6	**4.4**
Tedlium	10.5	**4.0**
CHiME6	65.8	**25.5**
VoxPopuli En	17.9	**7.3**
CORAAL	35.6	**16.2**
AMI IHM	37.0	**16.9**
Switchboard	28.3	**13.8**
CallHome	34.8	**17.6**
WSJ	7.7	**3.9**
AMI SDM1	67.6	**36.4**
LibriSpeech Other	6.2	**5.2**
Average	29.3	**12.8**

https://arxiv.org/abs/2212.04356

Motivation: NLP pretraining and audio



https://jalammar.github.io/illustrated-bert

Motivation: summary

- Unlabeled datasets size **surpasses** labeled datasets
- Audio foundational model: one **backbone** many tasks
- Model robustness
- Audio domain pretraining ~= **plain NLP** (BERT like) pretraining

Datasets: what unsupervised training requires

- **Acoustic variety:** noises, distortions, reverberations
- **Semantic variety:** speech domains (TED’s, movie dialogues, etc.)
- **Computational effectiveness:** how to handle long audios?
- **Language diversity:** how to gather data for low resource languages?

Datasets: common ground

Unsupervised:

- **GigaSpeech** – **YouTube** crawled multilingual dataset
- **VoxLingua107** – **YouTube** crawled multilingual dataset
- **VoxPopuli** – **European Parliament** (EP) event recordings

Supervised:

- **CommonVoice** – crowdsourced **multilingual** dataset, used **Wikipedia** texts
- **Librispeech** – audiobooks in **English**
- **FLEURS** – open sourced high quality **multilingual** dataset, recorded **Wikipedia** texts

Datasets: VoxLingua107

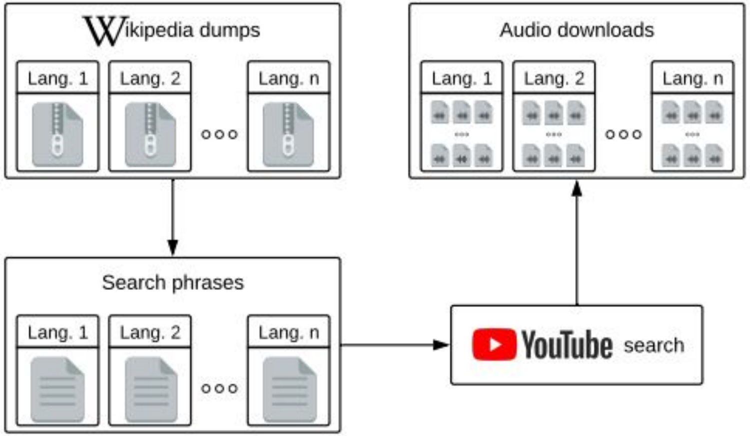


Fig. 1. High level overview of the data collection process.

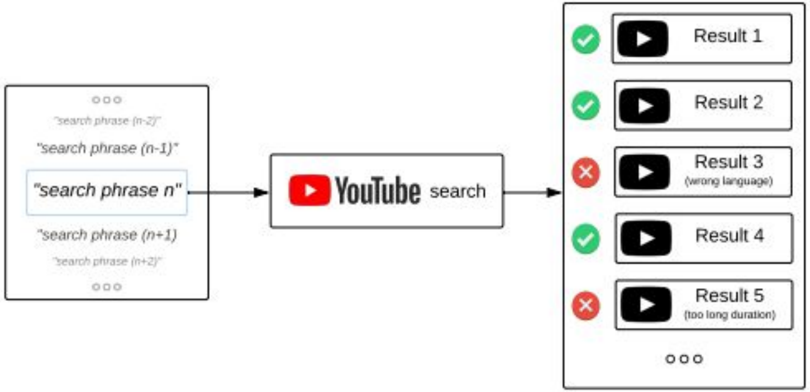
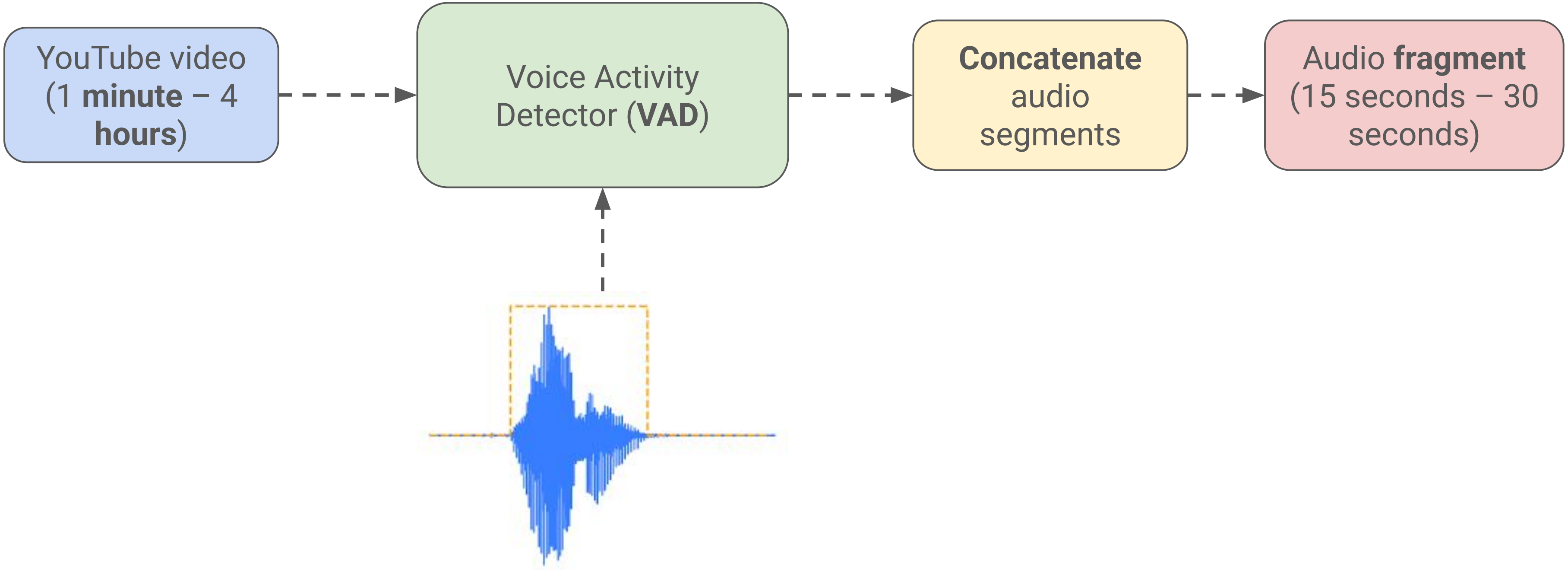


Fig. 2. Overview of the process of retrieving and filtering of videos.

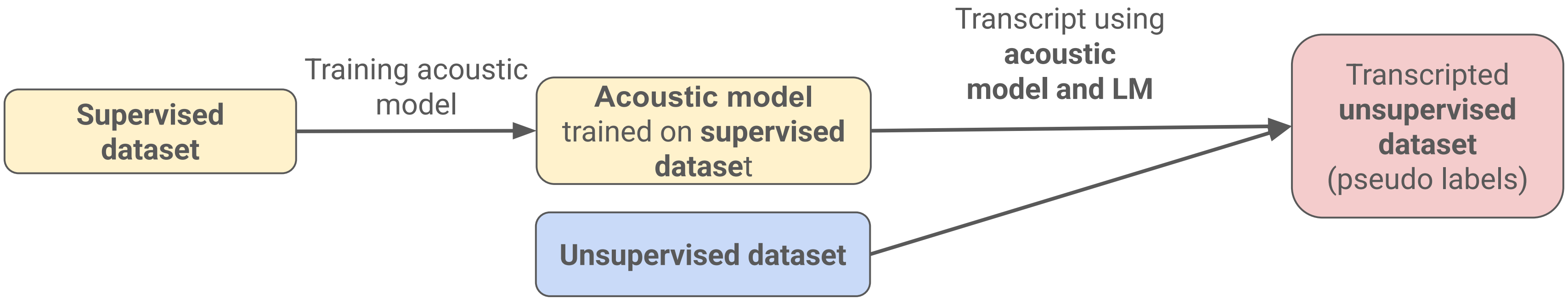
https://arxiv.org/abs/2011.12998

Datasets: audio fragment extraction

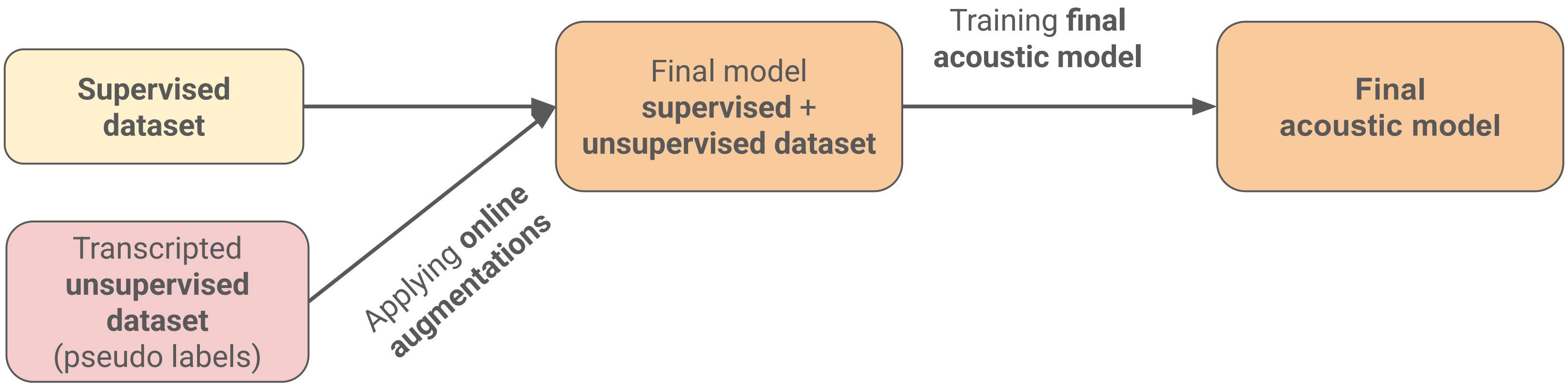


Self training: algorithm

Supervised training and decoding **(first stage)**



Training final model **(second stage)**



Self training: why it’s working?

- **Utilize external LM:** distilling knowledge of **AM + LM ensemble**
- **Online augmentations:** preventing final model being overconfident
- **Pseudo labels filtering:** drop over and under confident transcripts
- **“Statistical magic”**

Iterative pseudo labeling (IPL): idea

Algorithm 1: Iterative pseudo-labeling

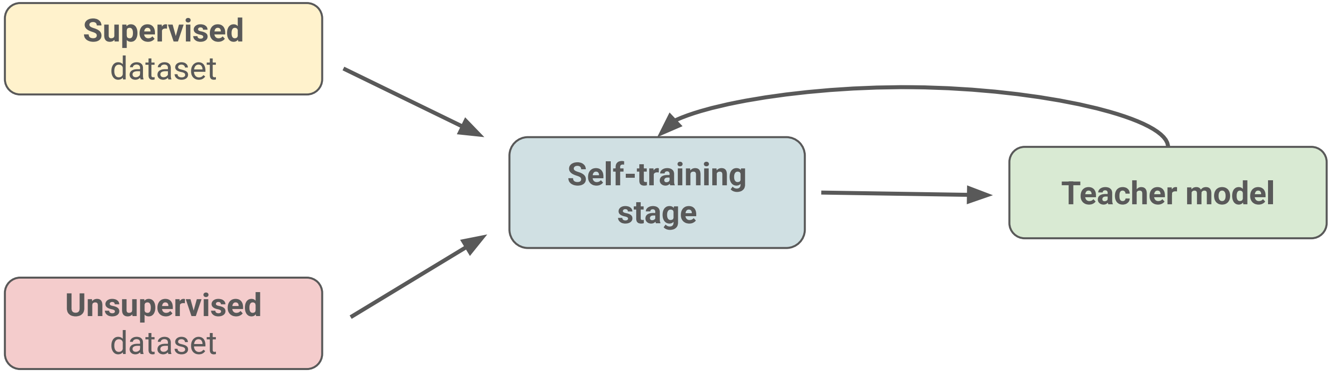
Data: Labeled data $L = \{x_i, y_i\}_{i=1}^{l}$, Unlabeled data $U = \{x'_j\}_{j=1}^{u}$
Result: Acoustic model p_θ
Initialize p_θ by training on only labeled data L;
repeat
  1. Draw a subset of unpaired data $\tilde{U} \in U$;
  2. Apply p_θ and decoding with LM to the subset $\tilde{U}$ to generate $\hat{U} = \{(x, \hat{y}) | x \in \tilde{U}\}$;
  3. Fine tune p_θ on $L \cup \hat{U}$ with data augmentation;
until *convergence or maximum iterations are reached*;

https://arxiv.org/abs/2005.09267

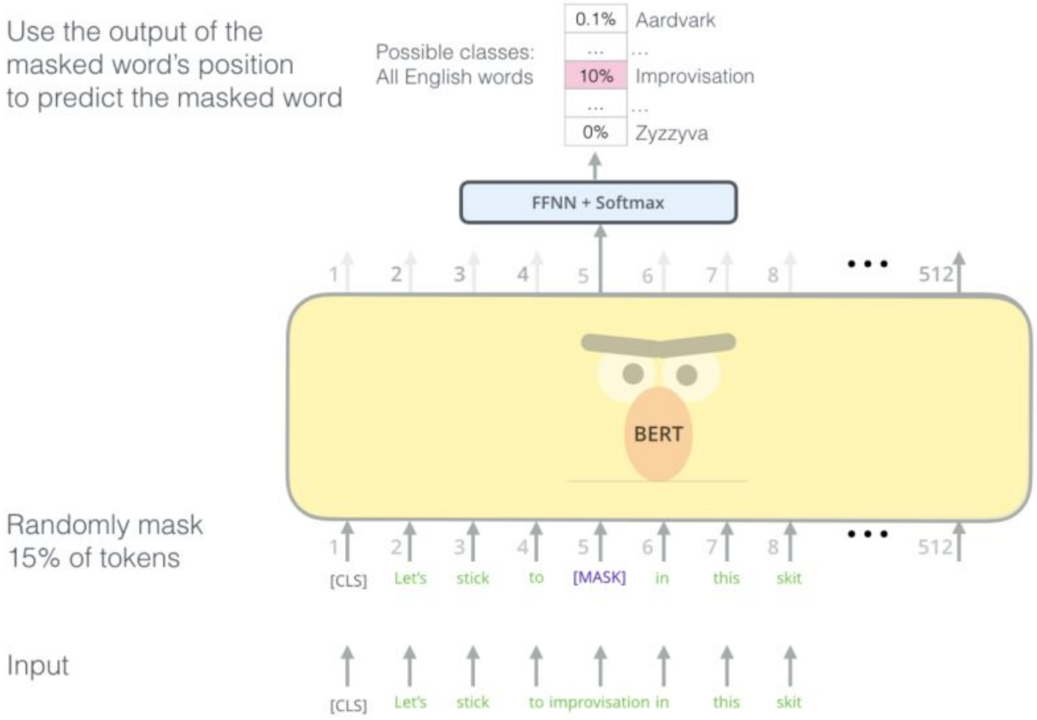
IPL

Supervised training and decoding (zero generation)

Iterative pseudo labeling process



BERT: recap

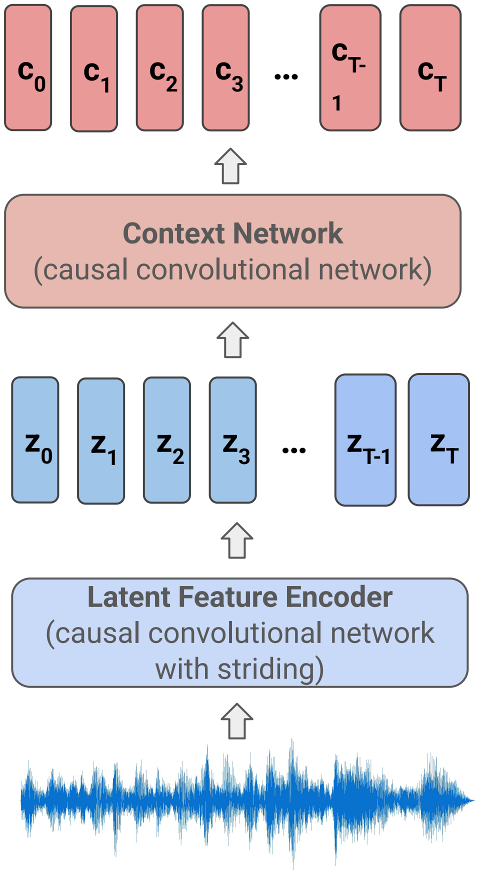


https://neptune.ai/blog/how-to-code-bert-using-pytorch-tutorial

Wav2Vec

Architecture overview

10ms original audio per $\mathbf{z}_i$



16K Hz waveform

Context vector

c_0

Mixed up latent representations

210ms – total receptive field for context vector

Latent space vector representation

z_i

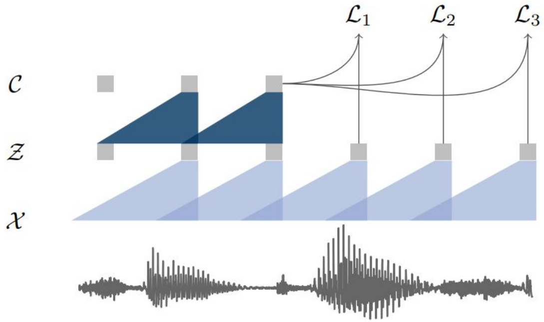
Low frequency audio vector representation

Audio frame encodes 30ms of original speech

https://arxiv.org/abs/1904.05862

Wav2Vec

Loss



k – **step** in future size

h_k – **affine transformation** for step k

$$\mathcal{L}_k = -\sum_{i=1}^{T-k} \left(\log \sigma(\mathbf{z}_{i+k}^\top h_k(\mathbf{c}_i)) + \lambda \mathop{\mathbb{E}}_{\tilde{\mathbf{z}} \sim p_n} [\log \sigma(-\tilde{\mathbf{z}}^\top h_k(\mathbf{c}_i))] \right)$$

Probability of latent z_{i+k} **being true**

Negative sampling from other audio positions (usually 10 distractors)

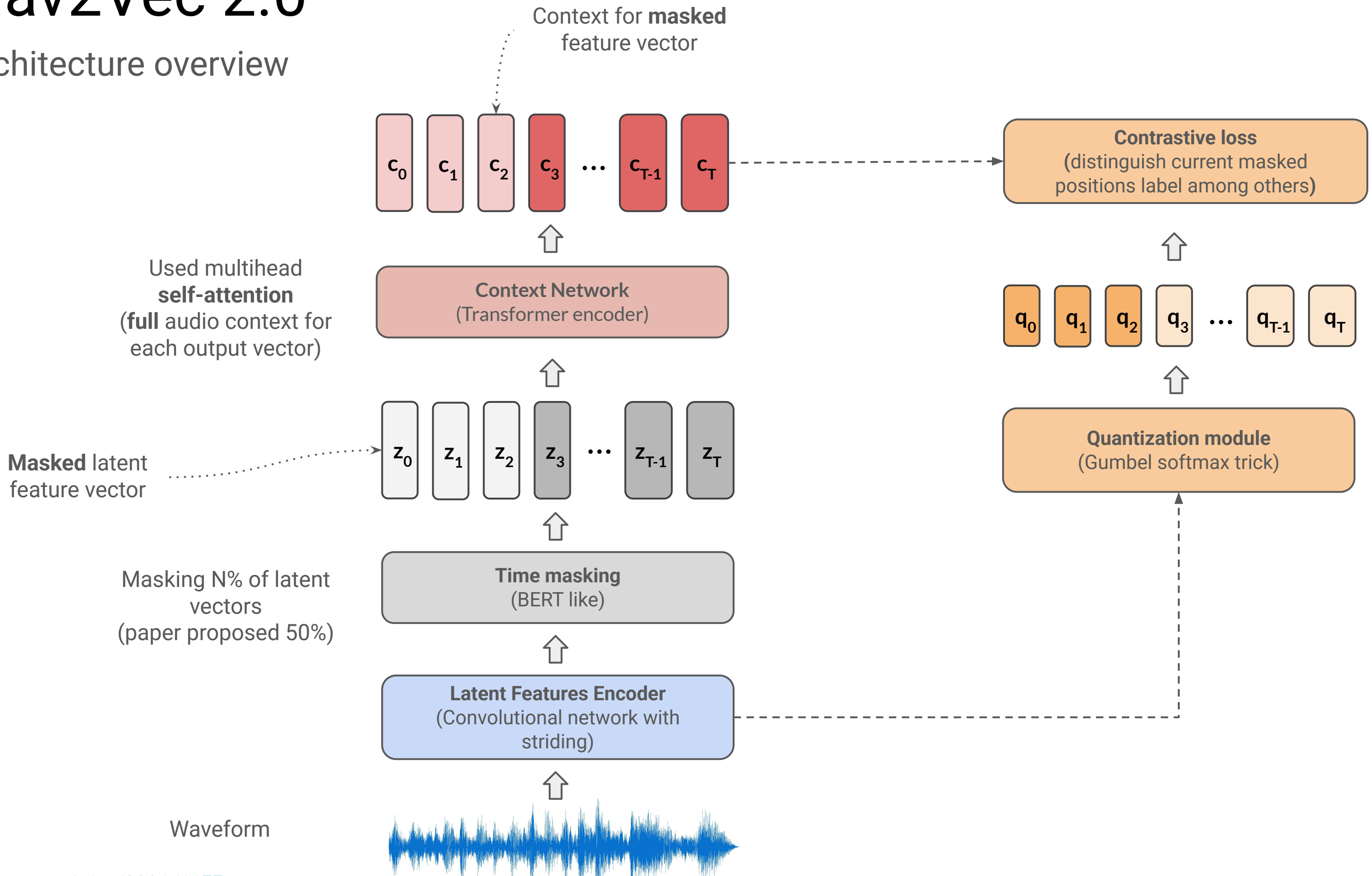
https://arxiv.org/abs/1904.05862

Wav2Vec: problems

- Causal context
- Step specific transform
- Why context vector should be closer to latent features?

Wav2Vec 2.0

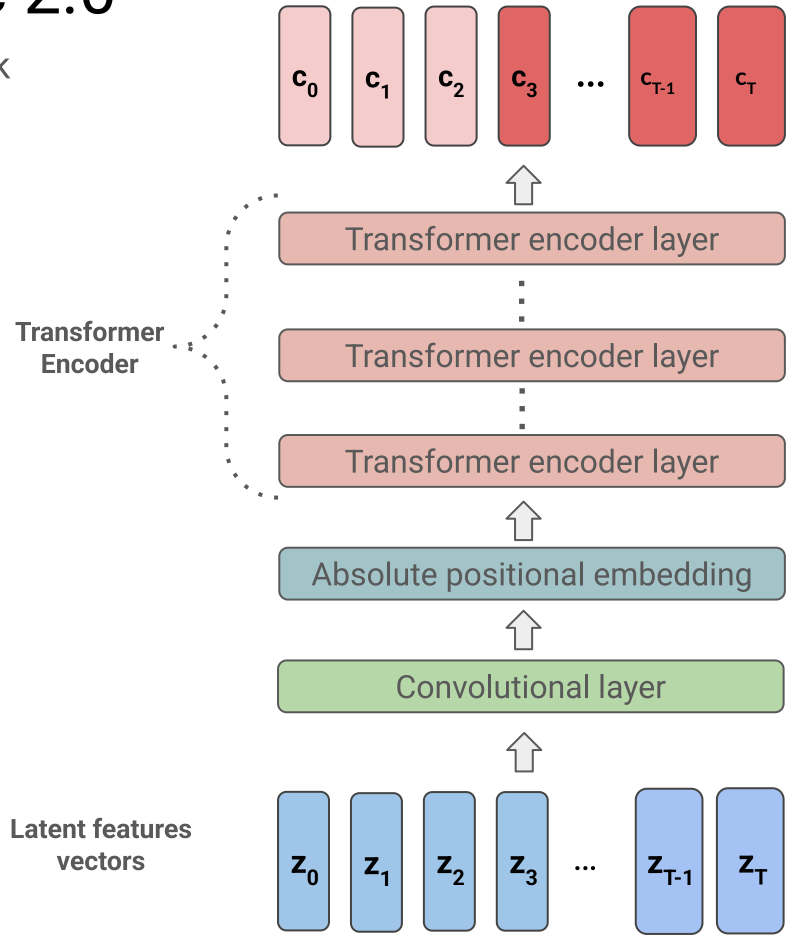
Architecture overview



https://arxiv.org/abs/2006.11477

Wav2Vec 2.0

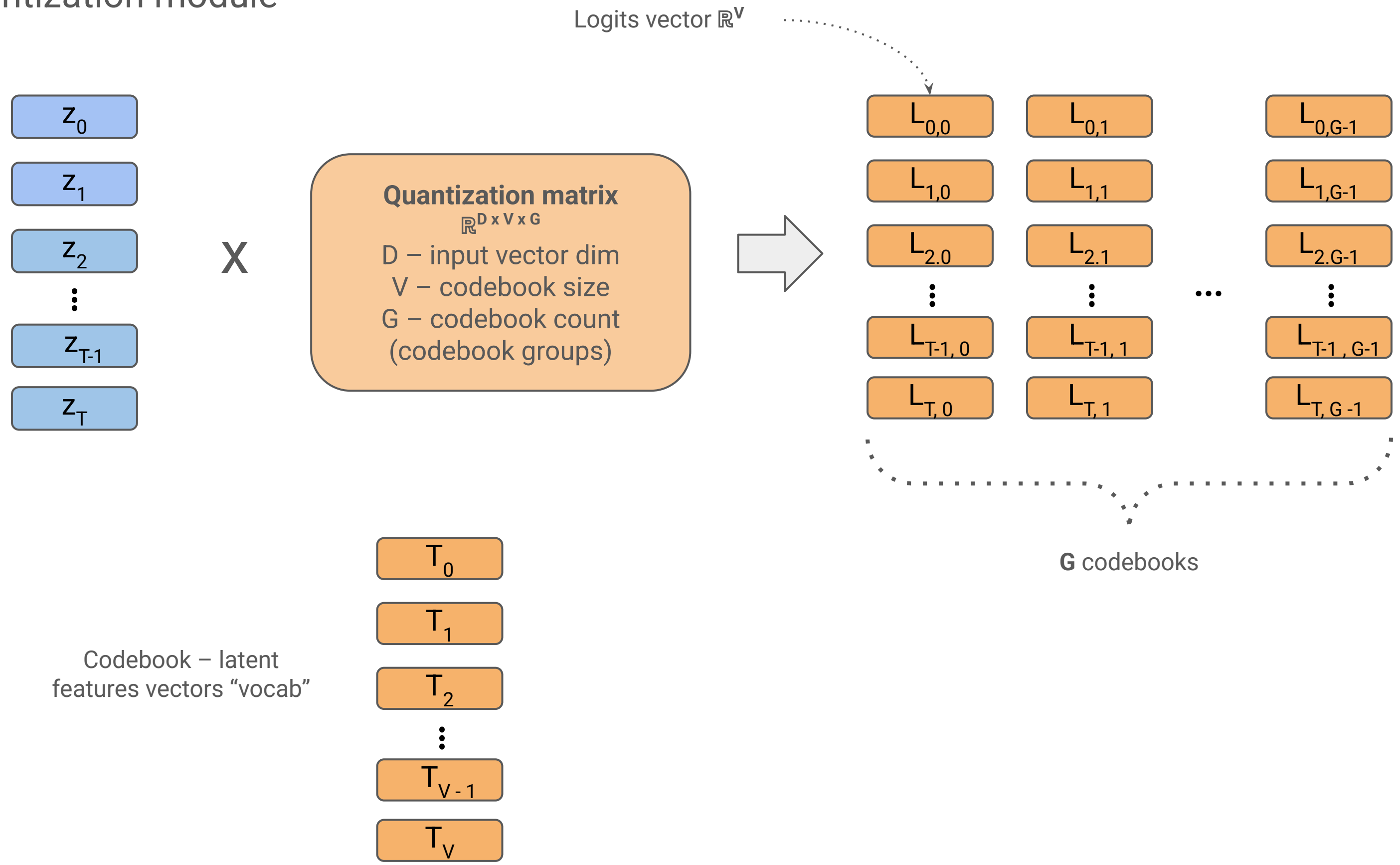
Context network



https://arxiv.org/abs/2006.11477

Wav2Vec 2.0

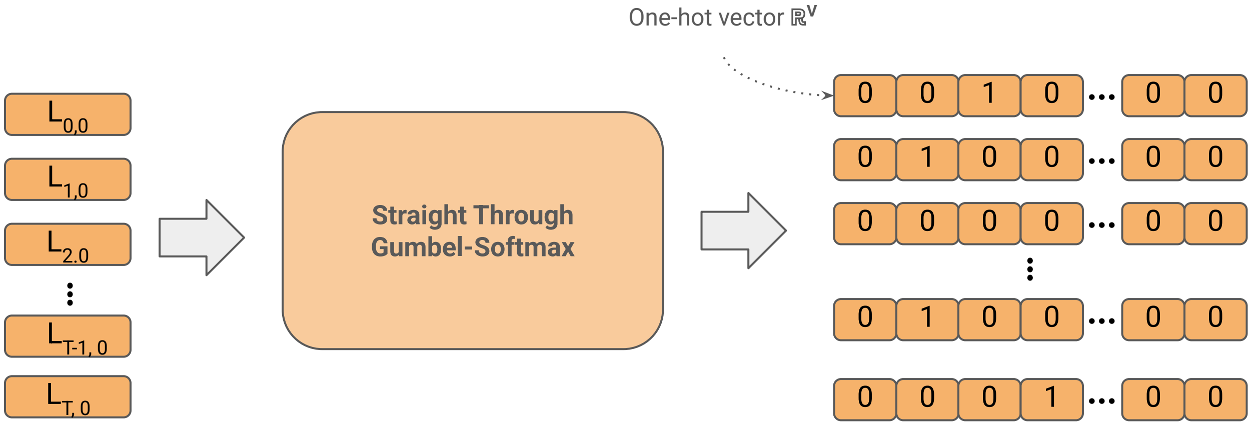
Quantization module



https://arxiv.org/abs/2006.11477

Wav2Vec 2.0

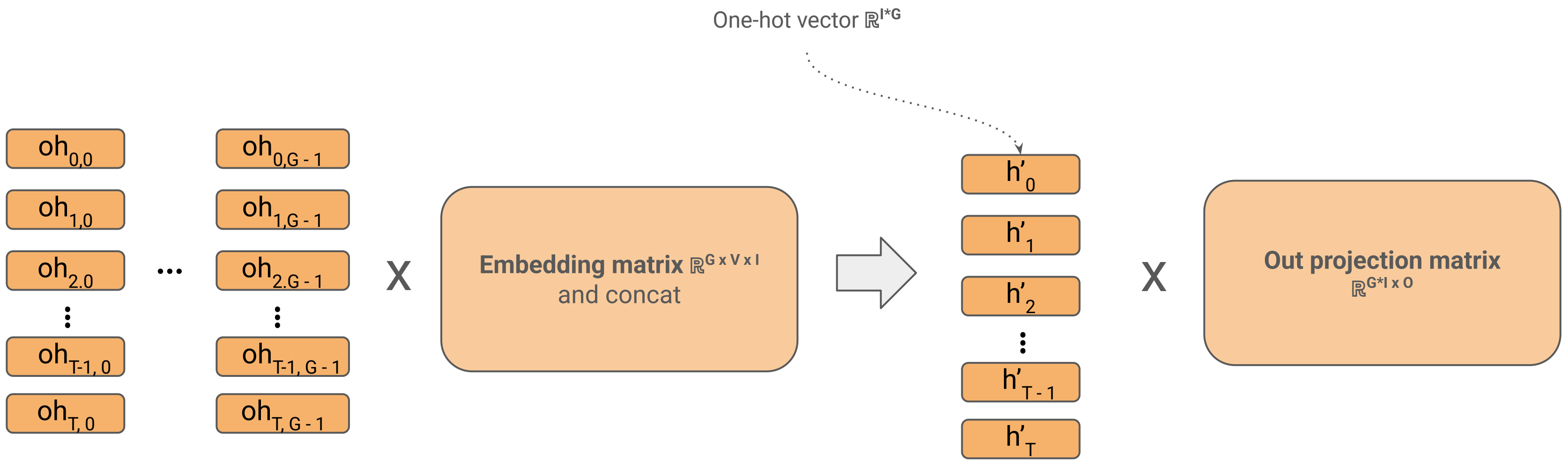
Quantization module



https://arxiv.org/abs/2006.11477

Wav2Vec 2.0

Quantization module



https://arxiv.org/abs/2006.11477

Wav2Vec 2.0

Sampling from categorical distribution

Parameter of distribution

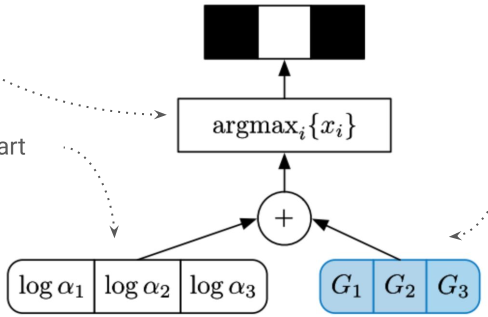
$$z = \texttt{one\_hot}\left(\arg\max_i \left[g_i + \log \pi_i\right]\right)$$

Formula to sample from categorical distribution

RV from Gumbel distribution

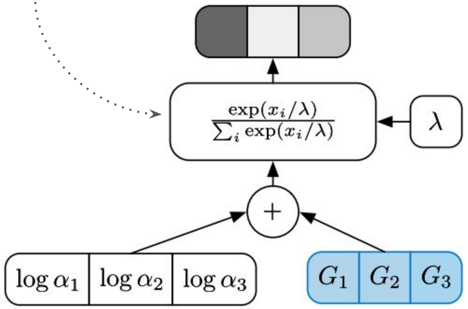
Non-differentiable

Parameters part



Stochastic part

Differentiable



https://arxiv.org/pdf/1611.01144v5

https://sassafras13.github.io/GumbelSoftmax/

Wav2Vec 2.0

Gumbel-Softmax trick

Logit of **v** token from **g** codebook

Sample from Gumbel distribution.
n = – log(– log(u)), u ~ U(0, 1)

Probability of **v** token from **g** codebook

Softmax temperature

$$p_{g,v} = \frac{exp\left[(l_{g,v} + n_v)/\tau\right]}{\sum_{k=1}^{V} exp\left[(l_{g,k} + n_k)/\tau\right]}$$

Straight through Gumbel-Softmax on forward pass

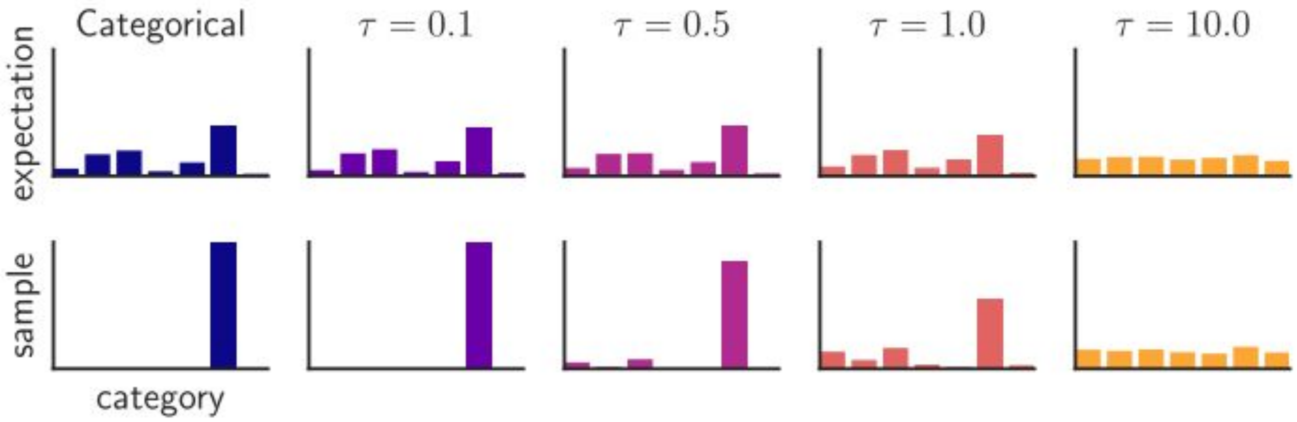
$$i = \text{argmax}_j p_{g,j}$$

Selected codebook token

https://arxiv.org/abs/2006.11477

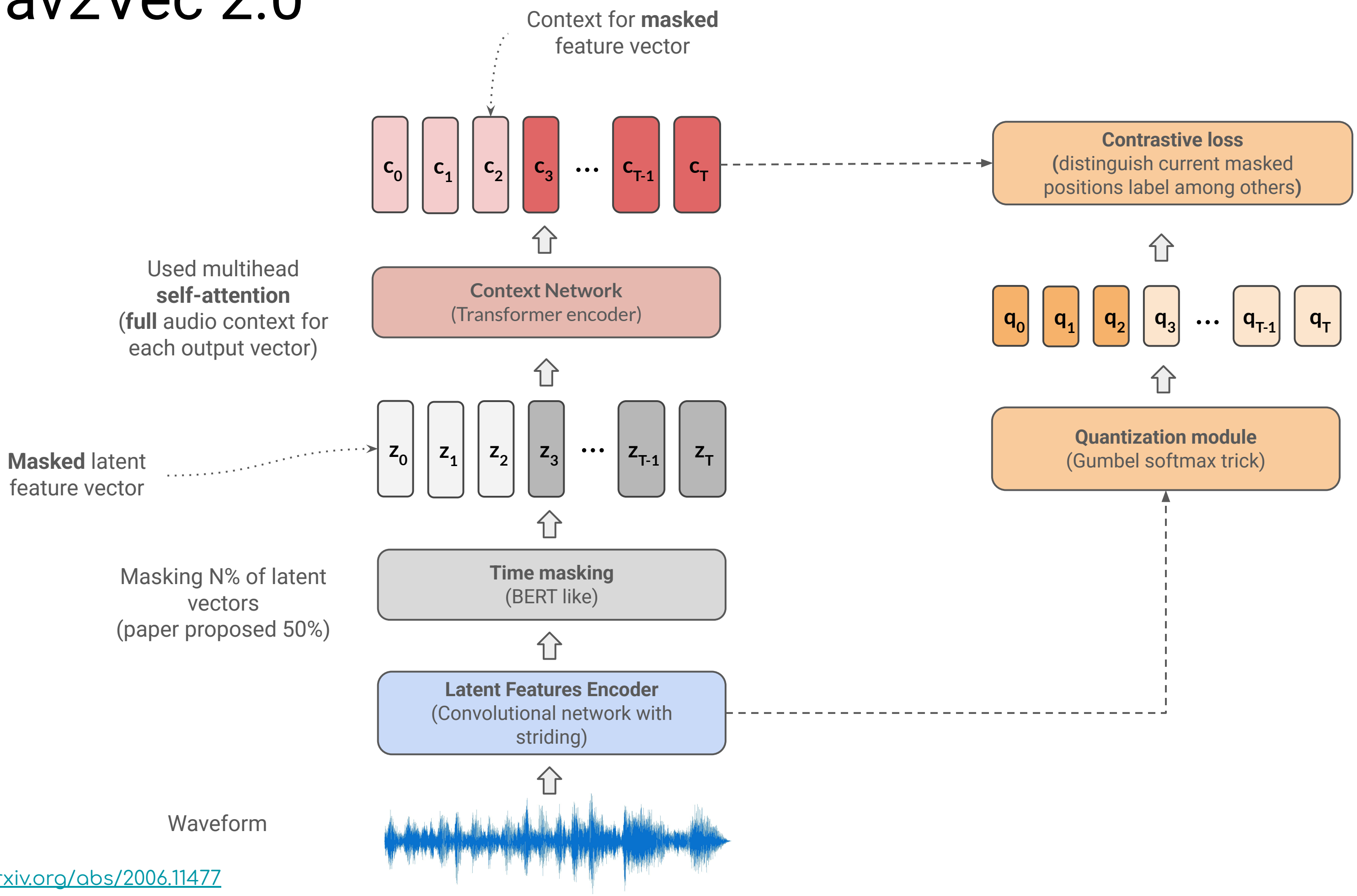
Wav2Vec 2.0

Gumbel-Softmax temperature



https://sassafras13.github.io/GumbelSoftmax/

Wav2Vec 2.0



https://arxiv.org/abs/2006.11477

Wav2Vec 2.0

Loss

Contrastive loss

Diversity loss

$$\mathcal{L} = \mathcal{L}_m + \alpha \mathcal{L}_d$$

Contrastive loss

Cosine similarity between context and quantized vectors

$$\mathcal{L}_m = -\log \frac{\exp(sim(\mathbf{c}_t, \mathbf{q}_t)/\kappa)}{\sum_{\tilde{\mathbf{q}} \sim \mathbf{Q}_t} \exp(sim(\mathbf{c}_t, \tilde{\mathbf{q}})/\kappa)}$$

Current timestamp (positive) and k – 1 sampled distractors (negative)

Diversity loss

$$\mathcal{L}_d = \frac{1}{GV} \sum_{g=1}^{G} -H(\bar{p}_g) = \frac{1}{GV} \sum_{g=1}^{G} \sum_{v=1}^{V} \bar{p}_{g,v} \log \bar{p}_{g,v}$$

G codebook entropy

https://arxiv.org/abs/2006.11477

Wav2Vec 2.0: improvements over Wav2Vec

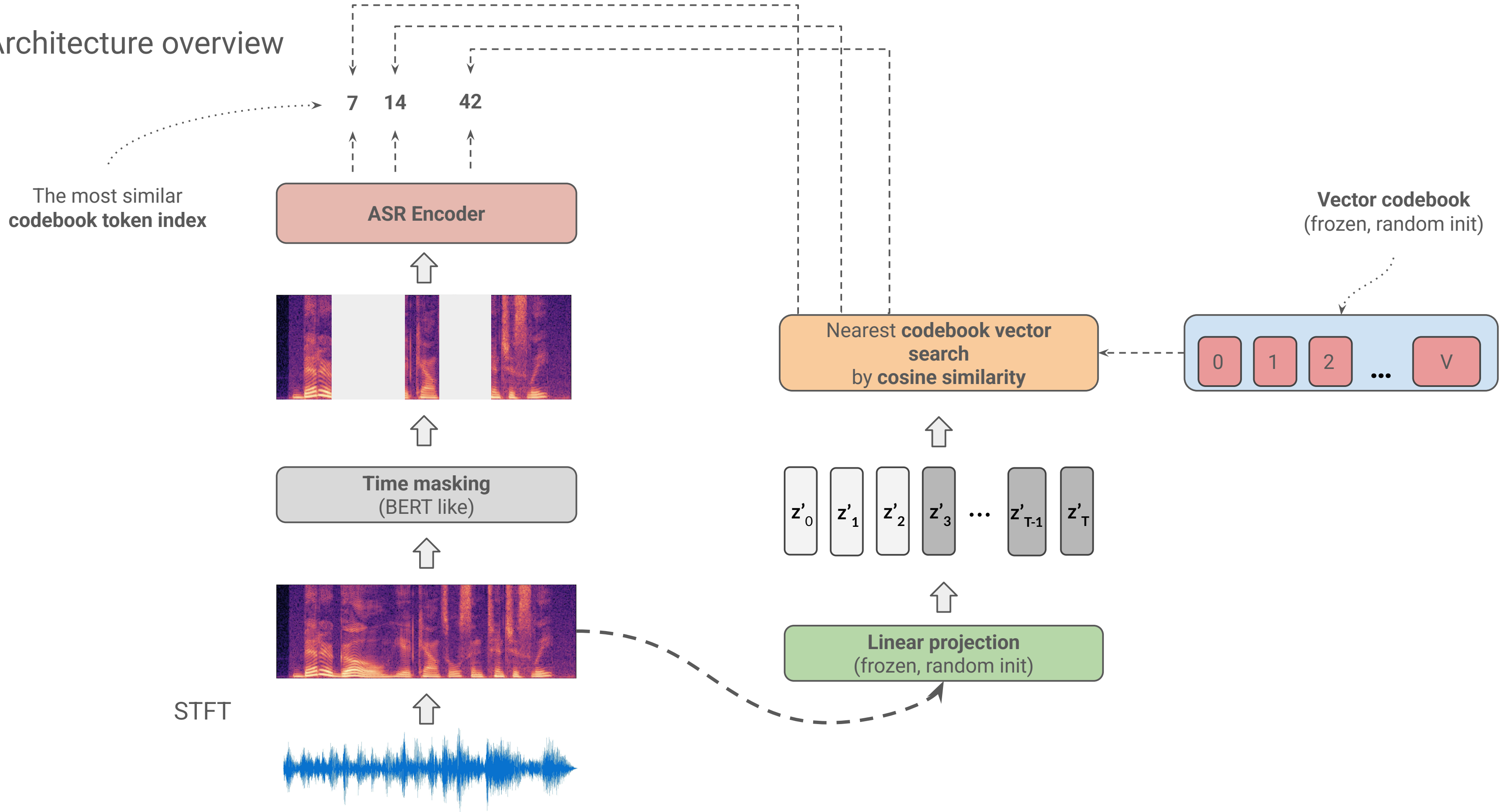
- Bidirectional context
- Quantization module allows retrieve more sophisticated targets

Wav2Vec 2.0: quantization problems

- Actor-critic or discriminator-generator problem
- Temperature scheduling
- Codebook interpretability (aka “audio” quantization)
- Codebook collapse

BEST-RQ

Architecture overview

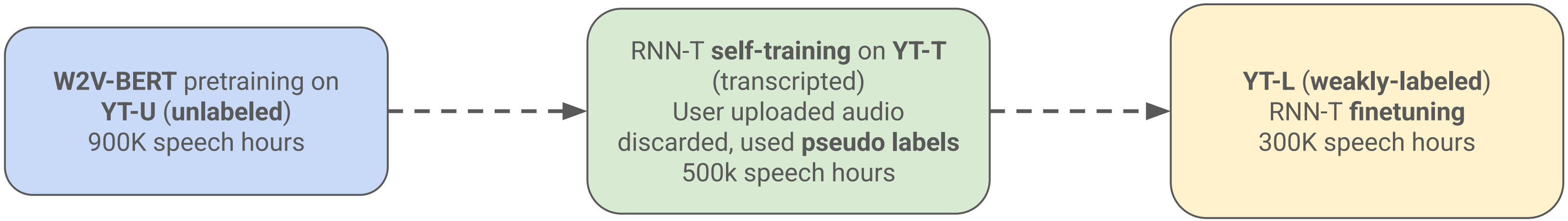


Pretraining and pseudo labeling for large scale modeling

Model scaling

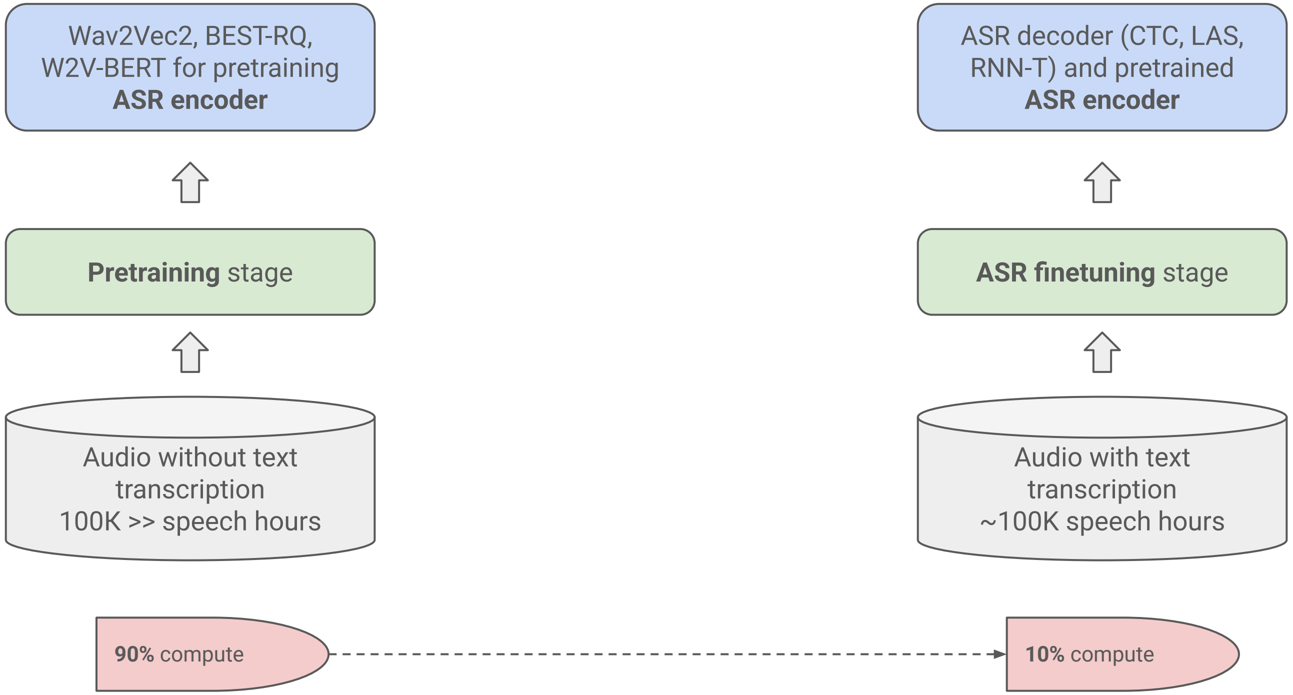
Model	# Params (B)	# Layers	Dimension	Att. Heads
Conformer XL	0.6	24	1024	8
Conformer XXL	1.0	42	1024	8
Conformer G	8.0	36	3072	16

Training pipeline



https://arxiv.org/abs/2109.13226
https://arxiv.org/abs/2010.10504

Pretraining and ASR finetuning



https://arxiv.org/abs/2303.01037

Pretraining evaluation: ASR tasks

Task	Multilingual Long-form ASR				Multidomain en-US	Multilingual ASR	
Dataset	YouTube			CORAAL	SpeechStew	FLEURS	
Langauges	en-US	18	73	en-US	en-US	62	102
Prior Work (single model)							
Whisper-longform	17.7	27.8	-	23.9	12.8		
Whisper-shortform[†]	-	-	-	13.2[‡]	11.5	36.6	-
Our Work (single model)							
USM-LAS	14.4	19.0	29.8	**11.2**	**10.5**	**12.5**	-
USM-CTC	**13.7**	**18.7**	**26.7**	12.1	10.8	15.5	-
Prior Work (in-domain fine-tuning)							
BigSSL [3]	14.8	-	-	-	7.5	-	-
Maestro [67]					7.2		
Maestro-U [67]							26.0 (8.7)
Our Work (in-domain fine-tuning)							
USM	13.2	-	-	-	7.4	13.5	19.2 (6.9)
USM-M	**12.5**	-	-	-	**7.0**	**11.8**	**17.4 (6.5)**
Our Work (frozen encoder)							
USM-M-adapter[§]	-	-	-	-	7.5	12.4	17.6 (6.7)