SCHOOL OF DATA ANALYSIS

ASR V: Recent Developments in ASR

Andrey Malinin

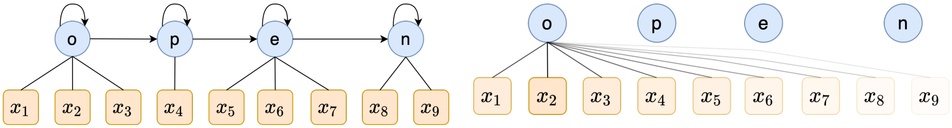
4th March 2022

Story so far

In this previous episode...

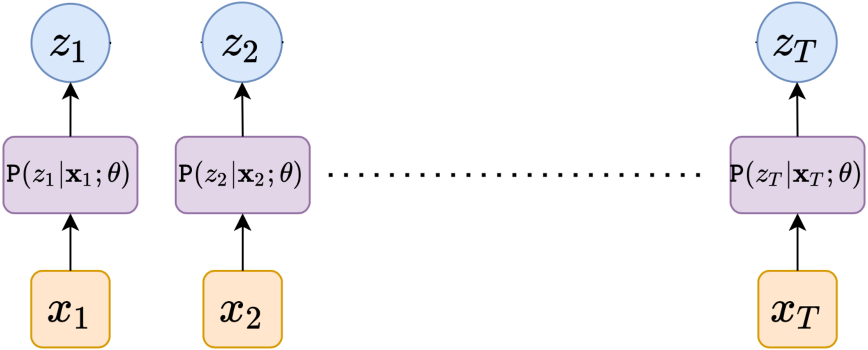
Recap - Data Processing and Alignment

- Process the Audio and Text into convenient representations
  - Transform audio into sequence of acoustic features or frames $\boldsymbol{X}_{1:T}$
  - Transform text into a sequence of speech units $\omega_{1:L}$
- Need to dynamically align features $\boldsymbol{X}_{1:T}$ to speech units $\boldsymbol{\omega}_{1:L} \rightarrow$ use:
  - State-Space models (HMMs and CTC)
  - Neural Attention Mechanisms



(a) State-Space (b) Attention Mechanism

Connectionist Temporal Classification (CTC)



Connectionist Temporal Classification

- Discriminative State-Space model $\rightarrow$ doesn't model inter-state dependencies

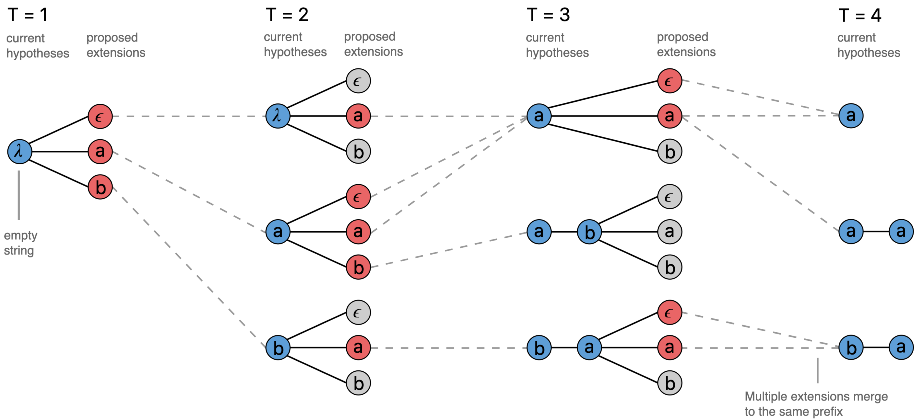
$$\mathrm{P}(\boldsymbol{z}_{1:T}|\boldsymbol{X}_{1:T}) = \prod_{t=1}^{T} \mathrm{P}(z_t|\boldsymbol{X}_{1:T})$$

- Independently take arg-maxes $\rightarrow$ yields most probable state sequence

$$\boldsymbol{\pi}^*_{1:T} = \arg\max_{\boldsymbol{\pi}_{1:T}} \prod_{t=1}^{T} \mathrm{P}(z_t = \pi_t|\boldsymbol{X}_{1:T})$$

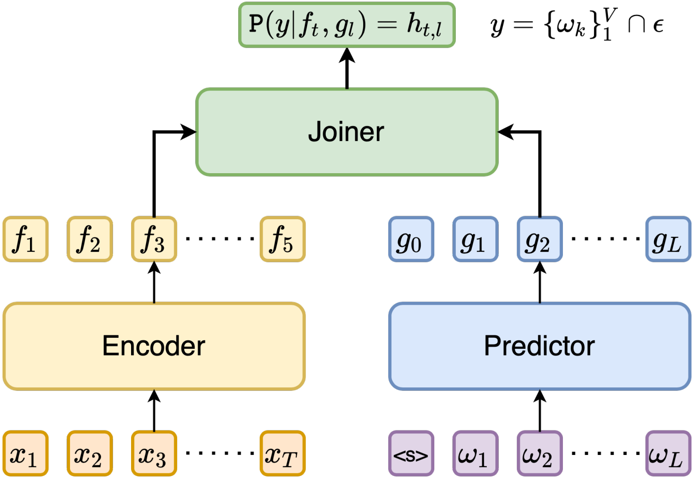
- GD can still fail to find the best solution $\rightarrow$
  - Grammatical constraints not enforced $\rightarrow$ output 'sounds', but has many errors.
- Use language model to enforce grammatic constraints!

Prefix Beam Search Decoding

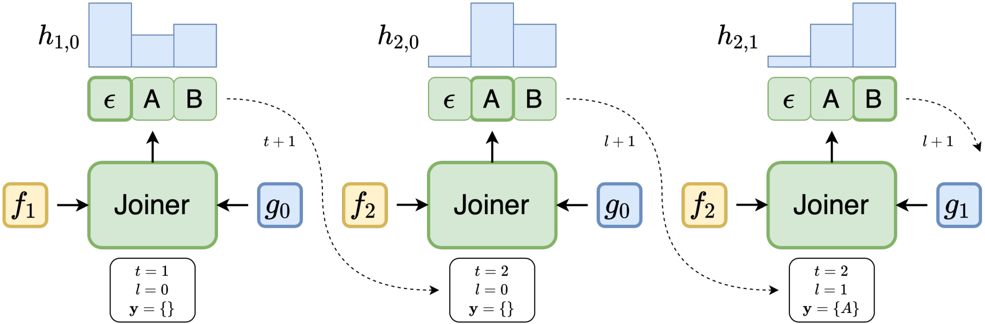


The CTC beam search algorithm with an output alphabet $\{\epsilon, a, b\}$ and a beam size of three.

RNN Transducer - Architecture

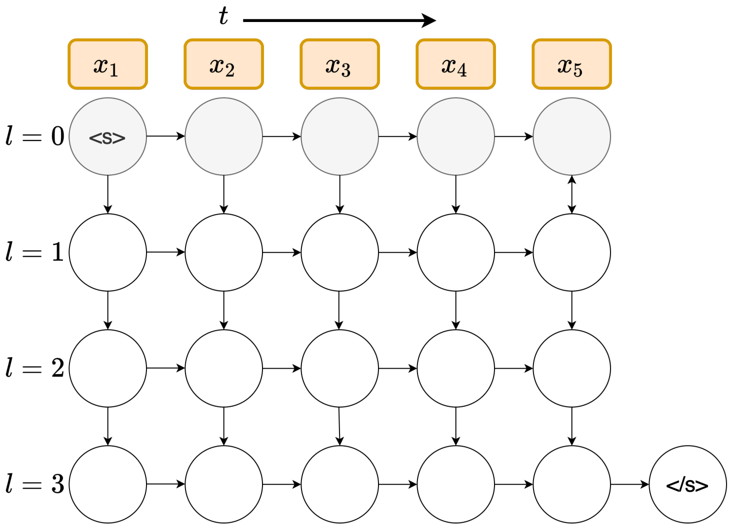


RNN Transducer Inference - Greedy Decoding



- Begin with empty prefix. Acoustic frame into $t = 1$, context index $l = 0$.
  - If ϵ is predicted $\rightarrow$ increment acoustic frame index t.
  - If character is predicted $\rightarrow$ increment l, append character to prefix.

RNN Transducer Alignment Trellis



Language Modelling

- A language model defines a prior distribution over word sequences:

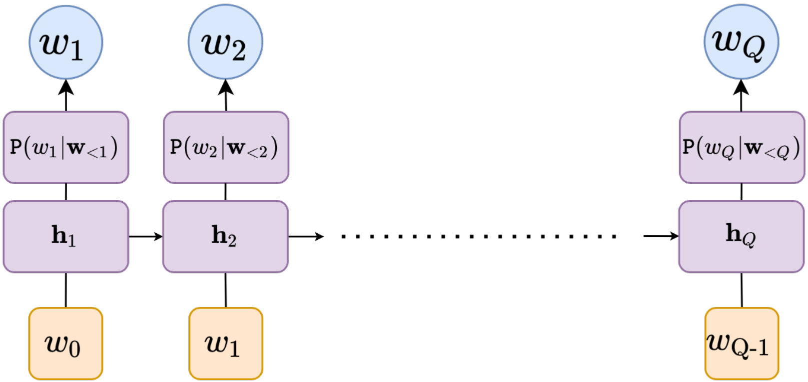
$$\mathrm{P}(\boldsymbol{w}) = \prod_{q=1}^{Q} \mathrm{P}(w_q | \boldsymbol{w}_{1:q-1})$$

- LMs help us discriminate between different acoustically plausible hypotheses:
  - Ex: "Wreck a nice beach" vs. "Recognize speech"
- LMs can also be defined at the character level or over BPE tokens
  - Choose appropriate context level based on task, language and amount of data
- Two common classes of language models:
  - N-Gram LMs $\rightarrow$ lightweight, cheap, limited flexibility
  - Neural LMs $\rightarrow$ expensive, powerful, expressive

- Language models are very useful for speech recognition!
  - Beam-search decoding
  - N-best list re-scoring
- **LM Fusion** - LMs can be combined with acoustic models on many levels
  - External LM
  - Integration of level of Neural Network architecture
- Choice of fusion level depends on architecture, language and data

Autoregressive Attention-based Models

Language Modelling - Neural Language Models (NLMs)



- NLMs express distribution over words as function of previous word and context

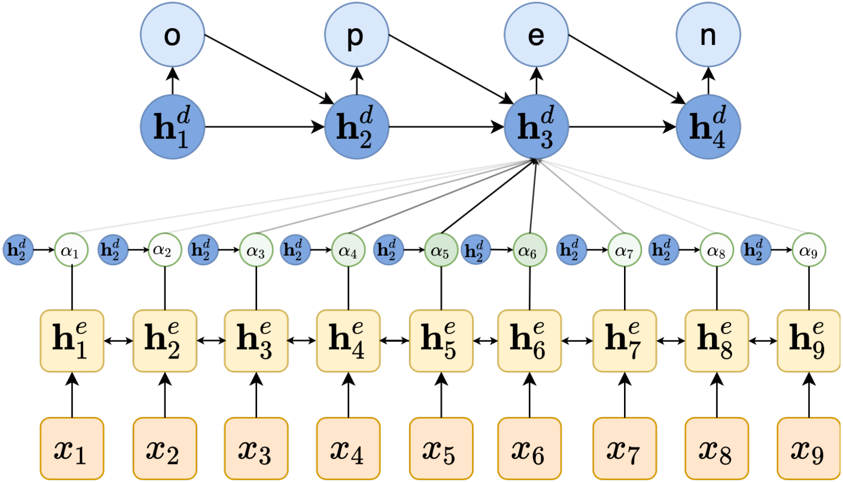
Autoregressive Attention-based Enoder-Decoder Models

- Autoregressive attention-based ASR $\rightarrow$ ASR via conditional LM

$$\mathrm{P}(\boldsymbol{w}_{1:Q}|\boldsymbol{X}_{1:T}) = \prod_{l=1}^{L} \mathrm{P}(w_q|\boldsymbol{w}_{<q}, \boldsymbol{X}_{1:T})$$

- Attention-based ASR systems have three main components
  - Module for generating text - the conditional NLM or Decoder
  - Module for processing and compressing audio - the Encoder
  - Module for aligning text and audio - Attention Mechanism
- Directly integrates LM and conditions on the acoustics
  - $+$ Jointly trains all components of ASR system
  - $+$ Mathematically simpler formulation (training, beam-search, alignment)
  - \- Needs MUCH more data.

Autoregressive Attention-based Models



Decoder

- The decoder is an NLM which generates text conditioned on the audio and history

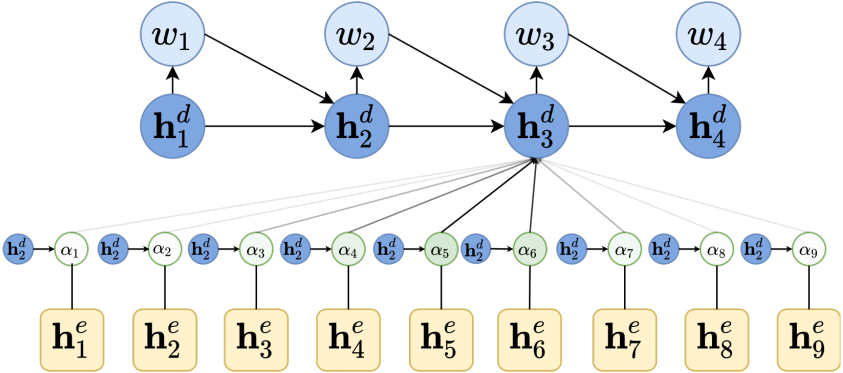
$$\mathrm{P}(\boldsymbol{w}_{1:Q}|\boldsymbol{X}_{1:T};\boldsymbol{\theta}) = \prod_{l=1}^{L} \mathrm{P}(w_q|\boldsymbol{w}_{<q},\boldsymbol{X}_{1:T};\boldsymbol{\theta}) = \prod_{l=1}^{L} \mathrm{P}(w_q|w_{q-1},\boldsymbol{h}_{q-1},\boldsymbol{c}_q;\boldsymbol{\theta}))$$

- Decoder is conditioned on previous word w_{q-1}, history $\boldsymbol{h}_{q-1}$ and audio-context $\boldsymbol{c}_q$
  - History vector $\boldsymbol{h}_{q-1}$ encodes the previously generated context
  - Audio-context $\boldsymbol{c}_q$ is a representation of $\boldsymbol{X}_{1:T}$ appropriate for generating next word
  - Audio-context $\boldsymbol{c}_q$ is provided by the attention mechanism and encoder
- Can generate a sentence either via sampling or beam-search

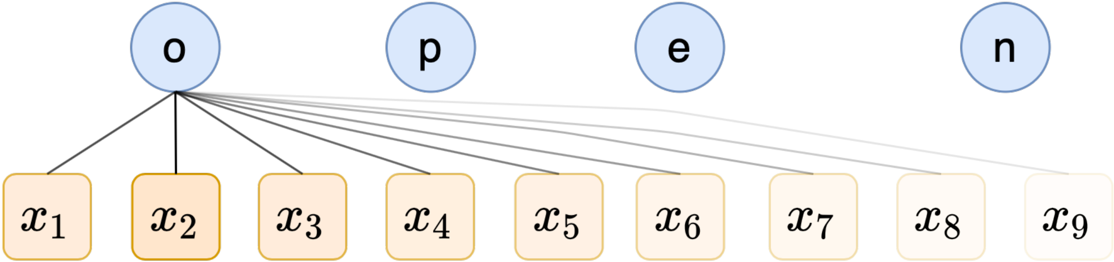
$$\boldsymbol{w}^*_{1:Q} = \arg\max_{\boldsymbol{w}_{1:Q}} \mathrm{P}(\boldsymbol{w}_{1:Q}|\boldsymbol{X}_{1:T};\boldsymbol{\theta})$$

- Training and Evaluation are mismatched!
  - Training - reference context. Evaluation - generated context!

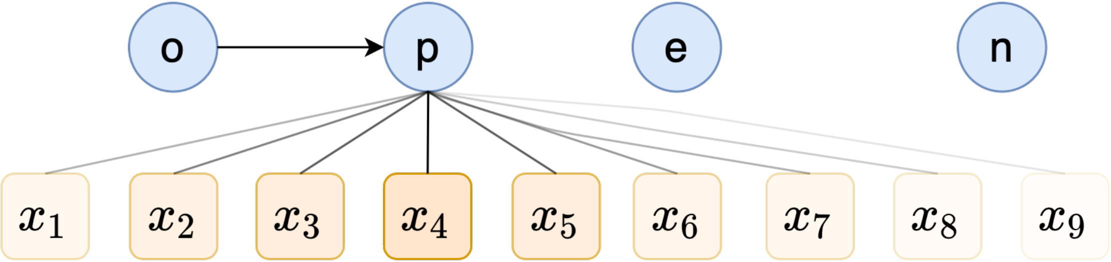
Autoregressive Attention-based Models



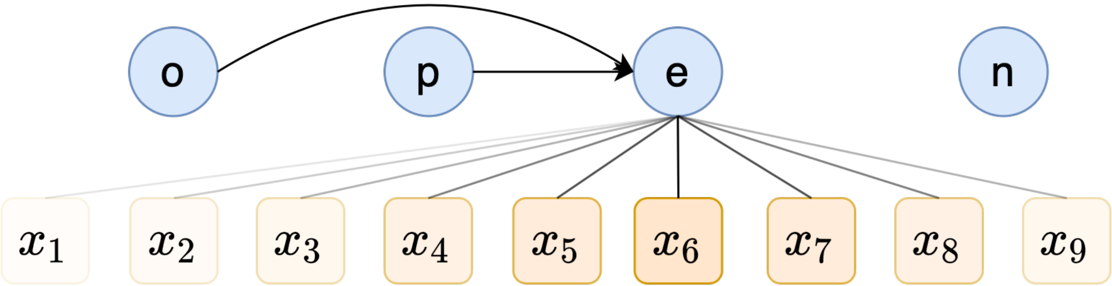
Alignment via Attention Mechanism



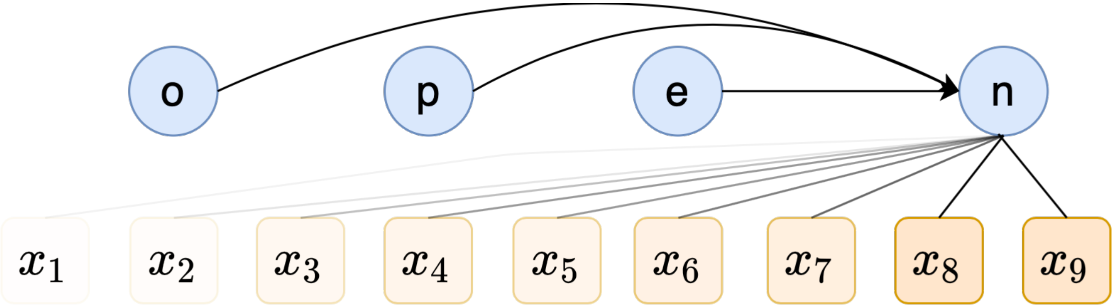
Alignment via Attention Mechanism



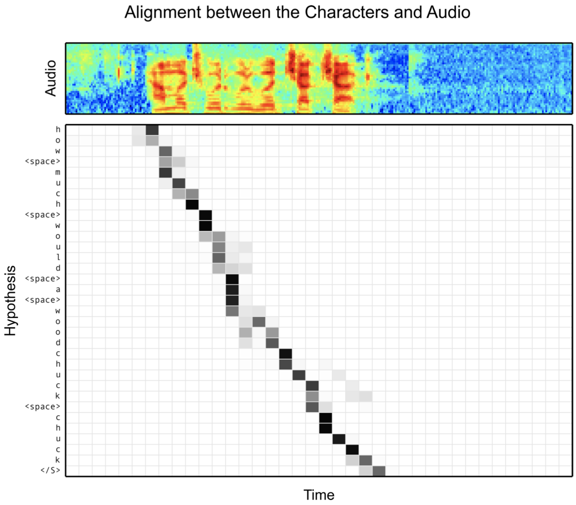
Alignment via Attention Mechanism



Alignment via Attention Mechanism



Attention-based Autoregressive Encoder-Decoder Models

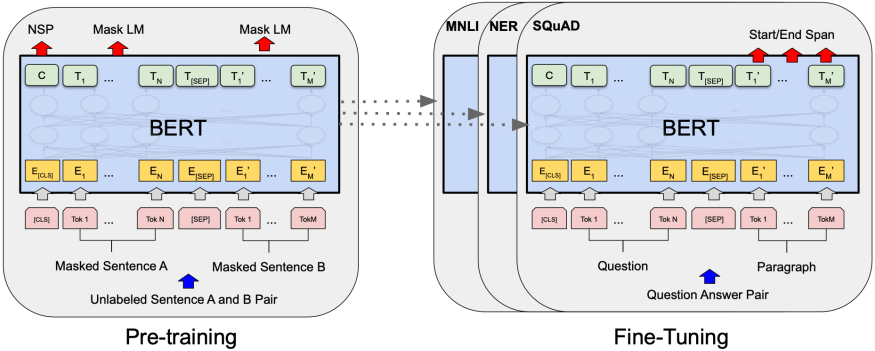


What's the best one can do with neural language models?

SOTA Language Modelling

- Language Modelling has exploded in recent years.
  - BERT, GPT2, GPT3, etc..
- Language models have been shown to be applicable to many tasks
  - Few-short learning
  - Solving NLU tasks (GLUE, Super GLUE)
  - Default 'pre-trained' model for NLP
- For ASR, they can be used in multiple ways, such as
  - N-Best list re-ranking
  - Pre-training architecture for semi-supervised learning
- Let's examine Masked Language Modelling

Masked Language Modelling - BERT



- BERT is a transformer-based Masked Language Model
  - Trained to predict masked words given seen context $\mathcal{L}(\boldsymbol{\theta}) = -\ln \mathrm{P}(w_{\text{masked}} | \boldsymbol{w}_{\text{seen}}; \boldsymbol{\theta})$

Semi-Supervised and unsupervised pre-training (Wav2vec, VQ-Wav2Vec)

How can we use un-labelled data?

Challenges of Attention-based Encoder-Decoder Models

- Best way to improve ASR performance?
  - Use more training data!
- However, manually labelling speech is very expensive and slow $\rightarrow$
  - Requires a pool of trained, professional annotators.
  - Crowd-sourcing provides noisy, potentially incorrect annotations.
  - Considerations regarding privacy.
- Can we somehow train ASR systems on unlabelled speech?
  - Yes! Use **semi-supervised learning**!
- Semi-Supervised Learning:
  - **Noisy-Student Training** (NST) and **Wav2Vec** (+ variations)

Semi-Supervised Learning

- Semi-supervised training leverages supervised $\mathcal{D}_\text{s}$ and unlabelled $\mathcal{D}_\text{u}$ data:

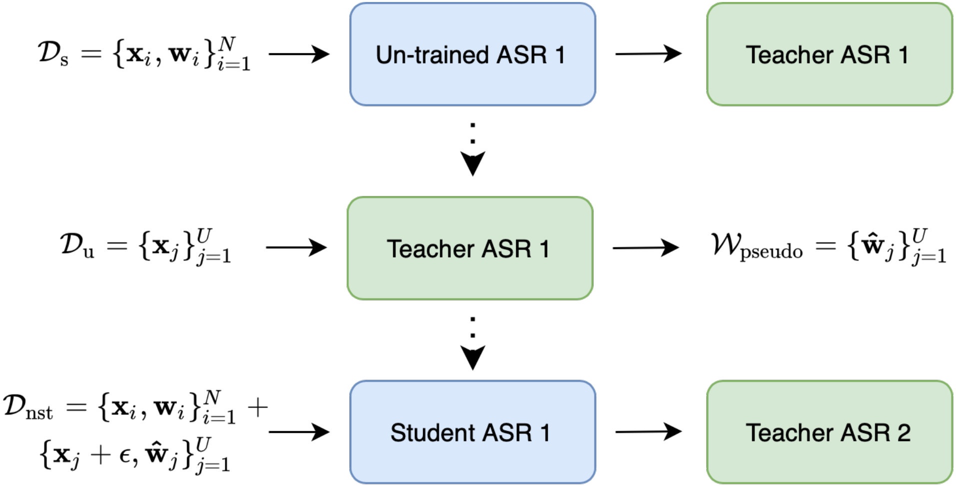
$$\mathcal{D}_\text{s} = \{\boldsymbol{x}_i, \boldsymbol{w}_i\}_{i=1}^{N}, \quad \mathcal{D}_\text{u} = \{\boldsymbol{x}_j\}_{j=1}^{U}$$

- Noisy-Student Training:
  - Uses a 'teacher' model trained on $\mathcal{D}_\text{s}$ to generate 'pseudo-labels' $\hat{\boldsymbol{w}}_{1:U}$ for $\mathcal{D}_\text{u}$
  - Train a new 'student' model both on supervised and pseudo-labelled data, adding noise (spec-augment) to $\mathcal{D}_\text{u}$
  - Re-label $\mathcal{D}_\text{u}$ using the new model

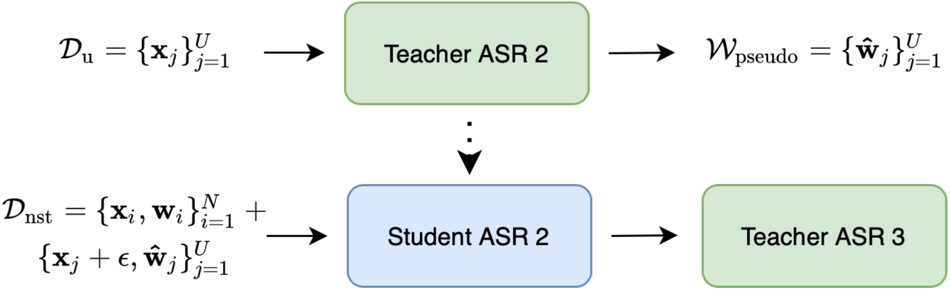
- Wav2Vec - Unsupervised Pre-training and Supervised Fine-Tuning
  - Use **contrastive learning** to train an encoder on all audio from $\mathcal{D}_\text{u}$ and $\mathcal{D}_\text{s}$
  - Finetune an ASR decoder on supervised data $\mathcal{D}_\text{s}$.

Noisy Student

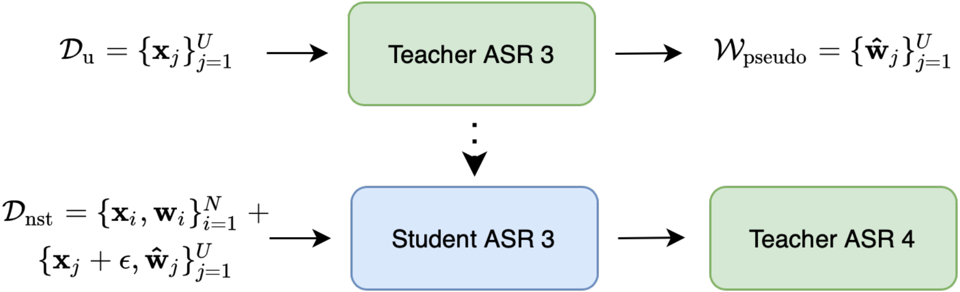
Noisy Student Training



Noisy Student Training



Noisy Student Training



Noisy Student Training

Method	Dev		Test	
	clean	**other**	**clean**	**other**
Supervised				
Lüscher et al., (2019) [39]	5.0	19.5	5.8	18.6
Kahn et al., (2019) [16]	7.78	28.15	8.06	30.44
Hsu et al., (2019) [19]	14.00	37.02	14.85	39.95
Ling et al., (2019) [31]			6.10	17.43
Semi-supervised (w/ LibriSpeech 860h)				
Kahn et al., (2019) [16]	5.41	18.95	5.79	20.11
Hsu et al., (2019) [19]	5.39	14.89	5.78	16.27
Ling et al., (2019) [31]			4.74	12.20
This Work				
Baseline (LAS + SpecAugment)	5.3	16.5	5.5	16.9
+ NST before LM Fusion	4.3	9.7	4.5	9.5
+ NST with LM Fusion	**3.9**	**8.8**	**4.2**	**8.6**

Noisy Student Training

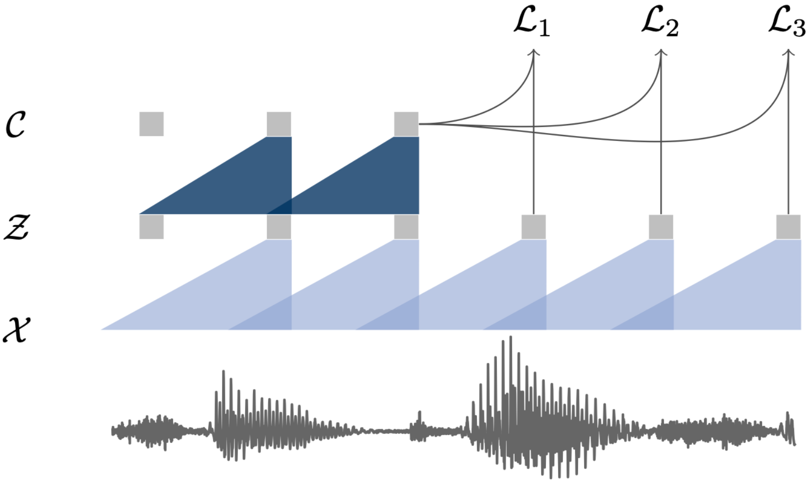
- NST works for several reasons:
  - Generates additional supervised training data with plausible 'pseudo-labels'
  - Smooths inputs around pseudo-labeled data via noise
  - Integrates knowledge form external LMs into ASR system
- NST can be improved via the following:
  - Use a more powerful language model
  - Filter out data which was badly pseudo-labelled
  - Do more iterations of pseudo-labelling
  - Use an ensemble of models throughout the process
- NST is a general technique which can be applied to other domains, such as vision.

Wav2Vec and it's variations

Wav2Vec

- Wav2Vec is an unsupervised data encoder
  - Wav2Vec operates directly on audio, not MelSpec
  - Wav2Vec doesn't need supervised training data
  - Trained via contrastric learning at fearture level
- ASR systems are trained on top of Wav2Vec using supervised data.
  - Can use any discriminative system, such as CTC, RNN-T or Seq2seq

Wav2Vec



$\mathcal{L}_1$
$\mathcal{L}_2$
$\mathcal{L}_3$
$\mathcal{C}$
$\mathcal{Z}$
$\mathcal{X}$

- Features are encoded using causal 1-D convolutions into representations $\boldsymbol{z}_{1:T}$.
  - Several layers of convolutions are used to reduce time-resolution
- A **context representation** $\boldsymbol{c}_{1:K}$ is computed.
- Model is trained to discriminate between representations K steps in the future and randomly chosen distractors

$$\mathcal{L}_k = -\sum_{i=1}^{T-k}\left(\ln\sigma(\boldsymbol{z}_{k+i}^{\mathrm{T}}\boldsymbol{h}_k(\boldsymbol{c}_i)) + \lambda\mathbb{E}_{\tilde{\boldsymbol{z}}\sim \mathrm{p}_n}\left[\ln\sigma(-\tilde{\boldsymbol{z}}^{\mathrm{T}}\boldsymbol{h}_k(\boldsymbol{c}_i))\right]\right)$$

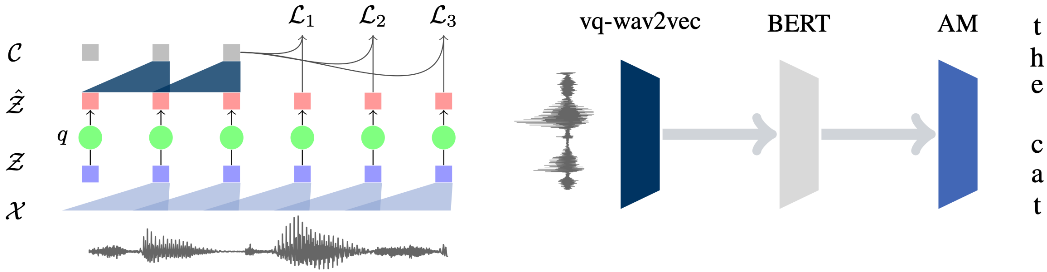
$$\boldsymbol{h}_k(\boldsymbol{c}_i) = \boldsymbol{W}_k\boldsymbol{c}_i + \boldsymbol{b}_k$$

$$\mathcal{L} = \sum_{k=1}^{K}\mathcal{L}_k$$

Wav2Vec

			nov93dev LER	nov93dev WER	nov92 LER	nov92 WER
Deep Speech 2 (12K h labeled speech; Amodei et al., 2016)			-	4.42	-	3.1
Trainable frontend (Zeghidour et al., 2018a)			-	6.8	-	3.5
Lattice-free MMI (Hadian et al., 2018)			-	5.66†	-	2.8†
Supervised transfer-learning (Ghahremani et al., 2017)			-	4.99†	-	2.53†
4-GRAM LM (Heafield et al., 2013)						
Baseline	–	–	3.32	8.57	2.19	5.64
wav2vec	Librispeech	80 h	3.71	9.11	2.17	5.55
wav2vec	Librispeech	960 h	2.85	7.40	1.76	4.57
wav2vec	Libri + WSJ	1,041 h	2.91	7.59	1.67	4.61
wav2vec large	Librispeech	960 h	2.73	6.96	1.57	4.32
WORD CONVLM (Zeghidour et al., 2018b)						
Baseline	–	–	2.57	6.27	1.51	3.60
wav2vec	Librispeech	960 h	2.22	5.39	1.25	2.87
wav2vec large	Librispeech	960 h	2.13	5.16	1.02	2.53
CHAR CONVLM (Likhomanenko et al., 2019)						
Baseline	–	–	2.77	6.67	1.53	3.46
wav2vec	Librispeech	960 h	2.14	5.31	1.15	2.78
wav2vec large	Librispeech	960 h	2.11	5.10	0.99	2.43

VQ-Wav2Vec



(a) vq-wav2vec

(b) Discretized speech training pipeline

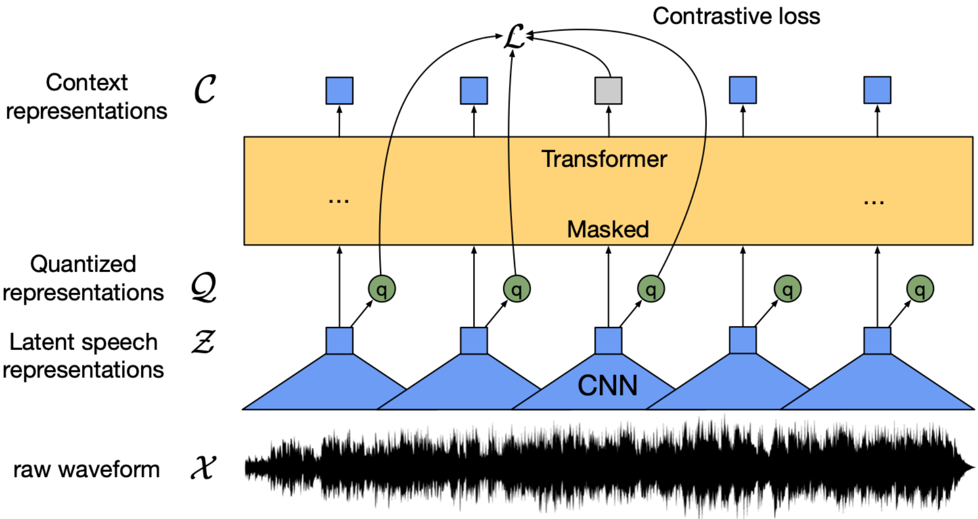
VQ-Wav2Vec

- Unlike Wav2Vec, VQ-Wav2Vec uses **quantized representations**
  - Helps to more efficiently compress relevant information
  - Allows learning **audio tokens**
  - Use Gumbel estimators to differentiate through discrete choice.
- Quantized audio representation are used to train a BERT-like model
  - Representations learn long-span context
- ASR system is trained on top of acoustic BERT embeddings

VQ-Wav2Vec

	nov93dev		nov92	
	LER	WER	LER	WER
Deep Speech 2 (12K h labeled speech; Amodei et al., 2016)	-	4.42	-	3.1
Trainable frontend (Zeghidour et al., 2018)	-	6.8	-	3.5
Lattice-free MMI (Hadian et al., 2018)	-	5.66†	-	2.8†
Supervised transfer-learning (Ghahremani et al., 2017)	-	4.99†	-	2.53†
No LM				
Baseline (log-mel)	6.28	19.46	4.14	13.93
wav2vec (Schneider et al., 2019)	5.07	16.24	3.26	11.20
vq-wav2vec Gumbel	7.04	20.44	4.51	14.67
+ BERT base	**4.13**	**13.40**	**2.62**	**9.39**
4-gram LM (Heafield et al., 2013)				
Baseline (log-mel)	3.32	8.57	2.19	5.64
wav2vec (Schneider et al., 2019)	2.73	6.96	1.57	4.32
vq-wav2vec Gumbel	3.93	9.55	2.40	6.10
+ BERT base	**2.41**	**6.28**	**1.26**	**3.62**
Char ConvLM (Likhomanenko et al., 2019)				
Baseline (log-mel)	2.77	6.67	1.53	3.46
wav2vec (Schneider et al., 2019)	2.11	5.10	0.99	2.43
vq-wav2vec Gumbel + BERT base	**1.79**	**4.46**	**0.93**	**2.34**

Wav2Vec 2.0


Contrastive loss
$\mathcal{L}$
Context representations
$\mathcal{C}$
Transformer
...
...
Masked
Quantized representations
$\mathcal{Q}$
q
q
q
q
q
Latent speech representations
$\mathcal{Z}$
CNN
raw waveform
$\mathcal{X}$

Goodbye!