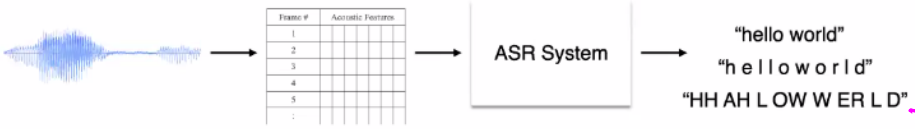
Day 7:

Speaker: Prof Preethi Jyothi

Title: Exploring the Landscape of Automatic Speech Recognition

What is Automatic Speech Recognition?

- Automatic speech recognition (ASR) systems: Accurately translate spoken utterances into text (words, syllables, etc.)
- Well-known examples: YouTube closed captioning, Voicemail transcription, Dictation systems, Siri front end, etc.



Speech Signal | Acoustic Features O | Transcription W

→ Phoneme (sound unit)

Why ASR?

- Speech is the primary means of human communication
- Develop natural interfaces for both literate & illiterate users
- Contribute to preservation of endangered languages → Indian Languages

What makes ASR a challenging problem?

Several sources of variability!

Style: Conversational (or casual) speech or read speech? Continuous speech or isolated words? Code-switched or monolingual?

Environment: Background noise, channel conditions, room acoustics, etc.

Speaker characteristics: Rate of speech, accent, age, etc.

Task specifics: Number of words in vocabulary, language constraints, etc. } → Repeated words ahh, hmm ... etc

How are ASR Systems Evaluated?

- Error rates computed on an unseen test set by comparing W* (predicted sentence) against W_{ref} (reference sentence) for each test utterance
  - Sentence/Utterance error rate (trivial to compute!)
  - Word/Phone error rate
- Word/Phone error rate (ER) uses the edit distance measure: What are the minimum number of edits (insertions/deletions/substitutions) required to convert W^* to W_{ref}?

Reference (W_{ref}): hello world

ASR Prediction (W^*): hell o world

Edit distance: 2

On a test set with N instances:

$$\text{ER} = \frac{\sum_{j=1}^{N} \text{Ins}_j + \text{Del}_j + \text{Sub}_j}{\sum_{j=1}^{N} \ell_j}$$

} minimize ER

Ins_j, Del_j, Sub_j are number of insertions/deletions/substitutions in the j^{th} ASR output

ℓ_j is the total number of words/phones in the j^{th} reference

Word Error Rates (WERs) Over the Years

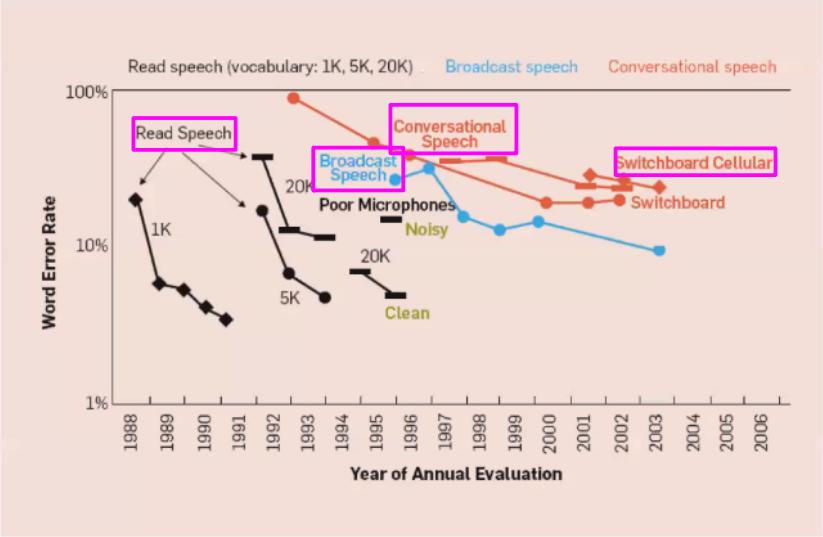


Figure from: "A Historical Perspective of Speech Recognition", Communications of the ACM, 2014

SwitchRate → ER ↑ ; Read Text in more controlled & clean text

but last decade

ER ↓

* 2006 → DL Models surpassed Hand Crafted Models

Progress in ASR Over the Last Decade

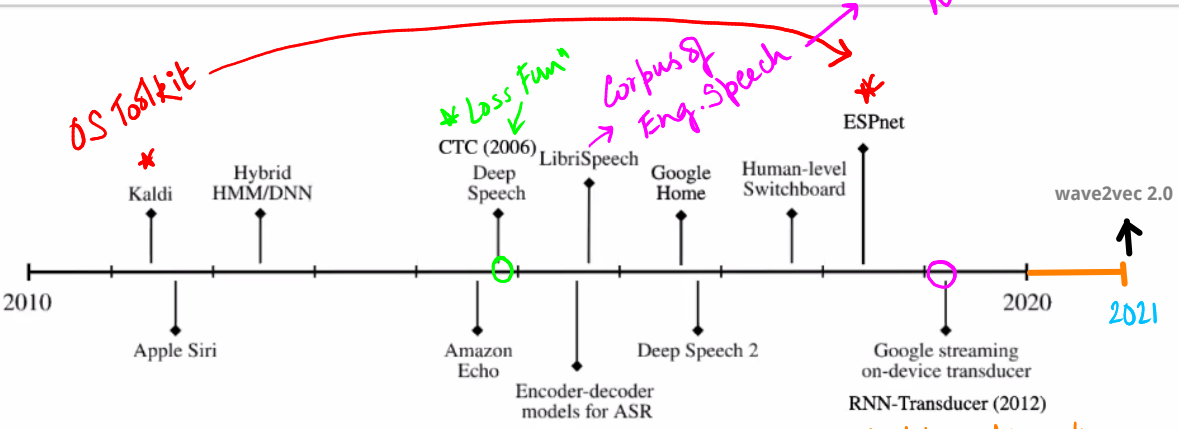


Figure from: "The History of Speech Recognition to the Year 2030" by Awni Hannun,https://awni.github.io/future-speech/

Hybrid HMM/DNN

Statistical Speech Recognition

Let $\mathbf{O}$ be a sequence of acoustic features corresponding to a speech signal. That is, $\mathbf{O} = \{O_1, \ldots, O_T\}$, where $O_i \in \mathbb{R}^d$ refers to a d-dimensional acoustic feature vector and T is the length of the sequence.

Let $\mathbf{W}$ denote a word sequence. An ASR system solves the following problem:

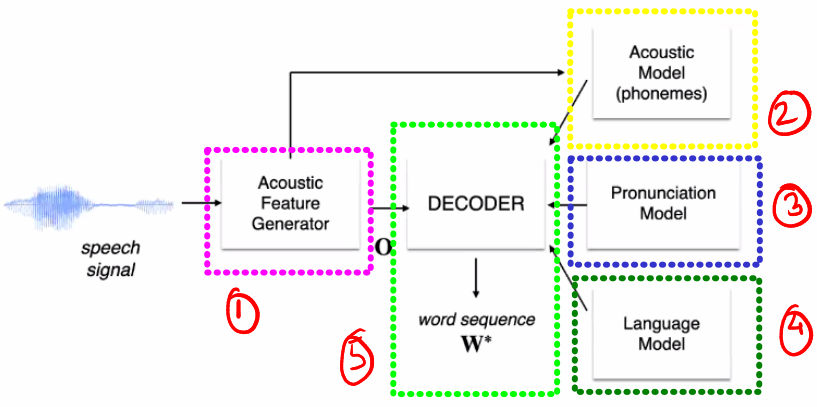
$$\begin{aligned}\mathbf{W}^* &= \arg\max_{W} \Pr(\mathbf{W} \mid \mathbf{O}) \\ &= \arg\max_{W} \Pr(\mathbf{O} \mid \mathbf{W}) \Pr(\mathbf{W}) \\ &\approx \arg\max_{W} \sum_{Q} \Pr(\mathbf{O} \mid \mathbf{Q}) \Pr(\mathbf{Q} \mid \mathbf{W}) \Pr(\mathbf{W})\end{aligned}$$

Language Model

Acoustic Model

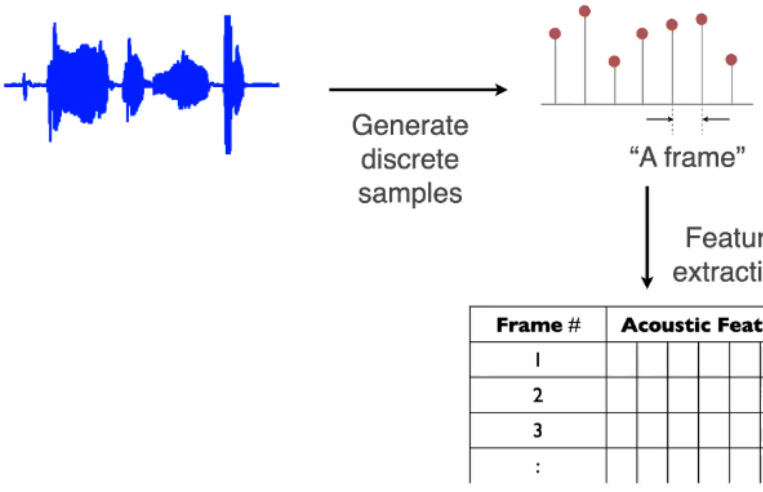
Pronunciation Model

Architecture of a Cascaded ASR system



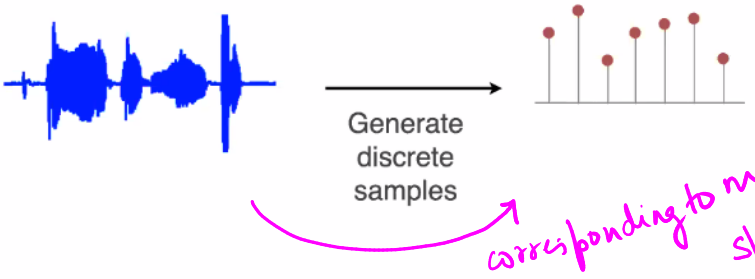
1

Speech Signal Analysis



Extract all the speech content

Speech Signal Analysis



corresponding to meaningful speech

- Need to focus on short segments of speech (*speech frames*) that more or less correspond to a phoneme and are stationary
- Each speech frame is typically 20-50 ms long
- Use overlapping frames with frame shift of around 10 ms
- Generate acoustic features corresponding to each speech frame

2

Basic Units of Acoustic Information

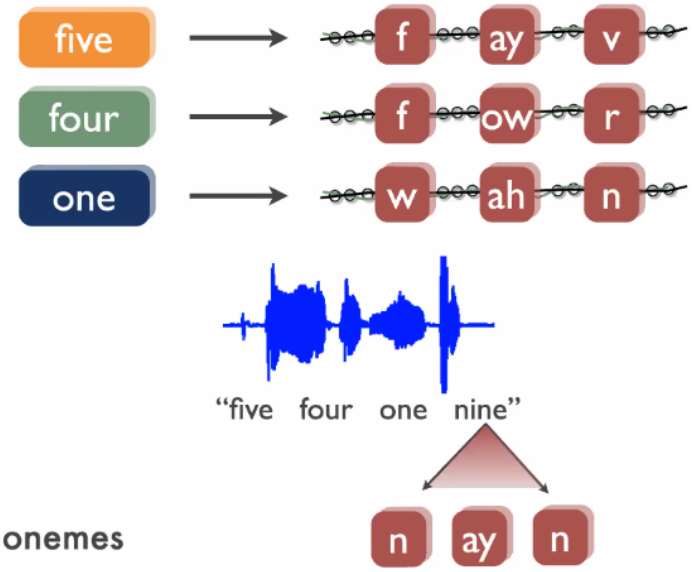
→ Why phonemes?

Phoneme: Abstract subword unit of speech that can be used to differentiate words

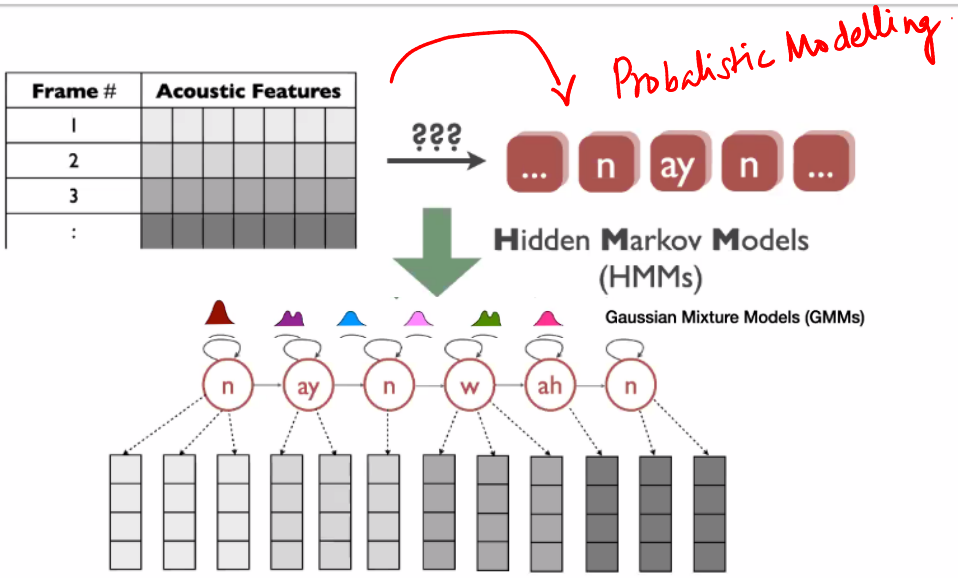
Minimal pair; e.g. "pan" vs. "can"

Most languages have 20-60 phonemes.

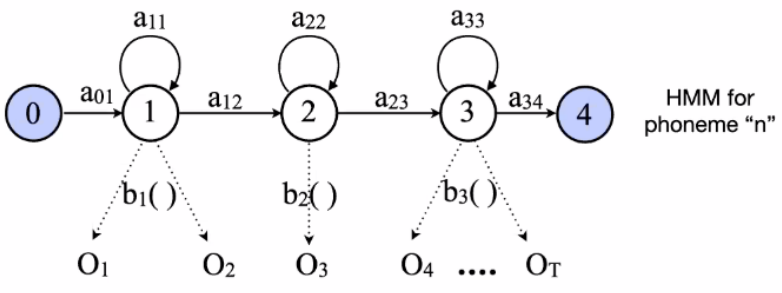
Why not use words as the basic unit?



Map from Acoustic Features to Phonemes



HMM for a Phoneme



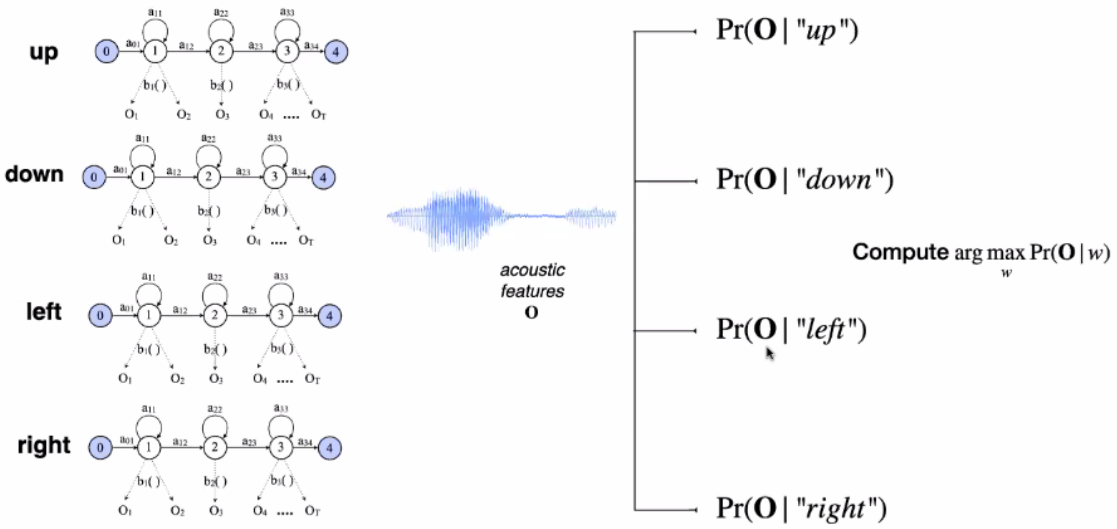
a_{ij} → Transition probabilities going from state i to state j

$b_j(O_i)$ → Probability of generating O_i from state j

Compute $\Pr(\mathbf{O} \mid \text{"n"}) = \sum_{\mathbf{Q}} \Pr(\mathbf{O}, \mathbf{Q} \mid \text{"n"})$

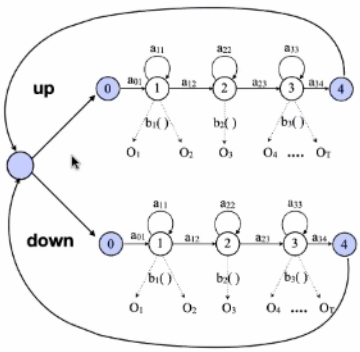
Compute using efficient "forward algorithm"

Isolated Word ASR



Small Tweak

- Task: Recognize utterances which consist of speakers saying either “up" or “down" **multiple times** per recording.

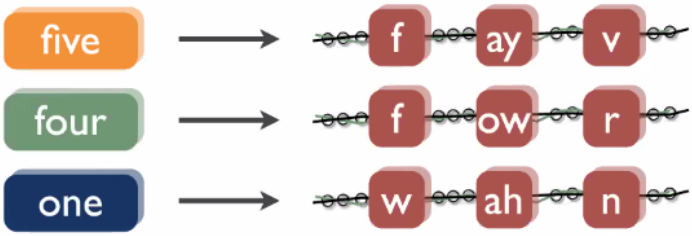


Combined HMM

Pronunciation Models

Provides link between phonemes and the word.

Typically, a simple dictionary of pronunciations is maintained.



Pronunciation Dictionary or Lexicon

- Pronunciation model/dictionary/lexicon: Lists one or more pronunciations for a word
- Typically derived from language experts: Sequence of phones written down for each word
- Dictionary construction involves:
  1. Selecting what words to include in the dictionary
  2. Pronunciation of each word (also, check for multiple pronunciations)

Handling Pronunciations: Some Challenges

- Same word can have multiple pronunciations
  - Accent: E.g., *route* /R UW T/ vs. /R AW T/
  - Part-of-speech: E.g., *conduct* (*verb*) /C UH N D AH K T/ vs. *conduct* (*noun*) /C AA N D AH K T/
  - Conversational effects: E.g., *probably* /P R AA B L IY/
- Most dictionaries only have words with a single pronunciation

4

Language Model

- Statistical language models → How likely is it for different words to occur given the recent word context?
  - For example, "the dog" ran?
    can?
    pan?
- Also useful to disambiguate between similar acoustics:
  - "Is the baby crying" vs. "Is the bay bee crying"
  - "Let us pray" vs. "Lettuce spray"

Language Models

- Ngram models: Limited memory of previous word history, only last m words are included
- 1-order language model (or bigram model)

$$\Pr(w_1,w_2,\ldots,w_{n-1},w_n) \cong \Pr(w_1|<s>)\ \Pr(w_2|w_1)\ \Pr(w_3|w_2)\ldots\Pr(w_n|w_{n-1})$$

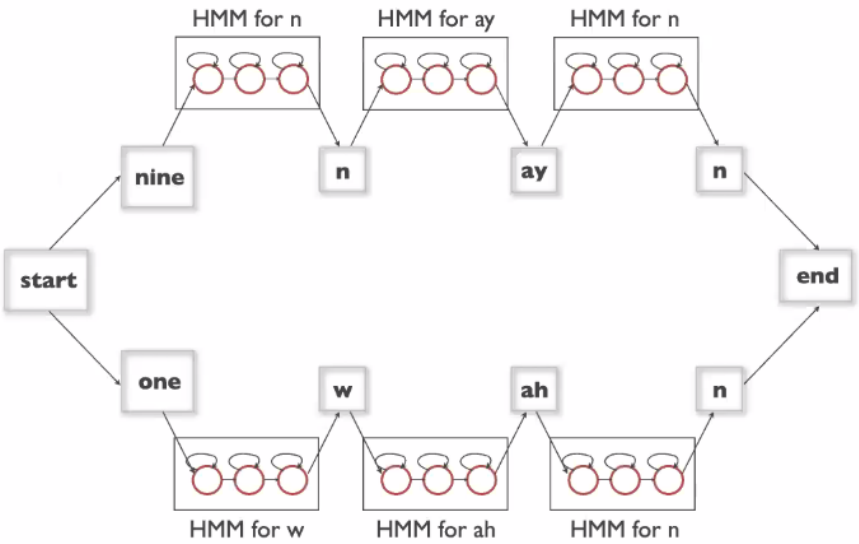
- 2-order language model (or trigram model)

$$\Pr(w_1,w_2,\ldots,w_{n-1},w_n) \cong \Pr(w_2|w_1,<s>)\ \Pr(w_3|w_1,w_2)\ldots\Pr(w_n|w_{n-2},w_{n-1})$$

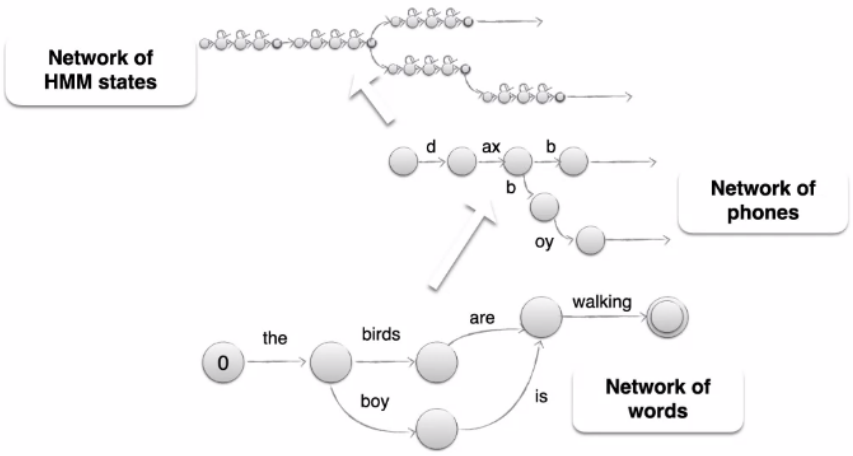
- Recurrent neural network-based (RNN) language models: Can model longer dependencies and use distributed word representations or *embeddings* for improved parameter sharing

5

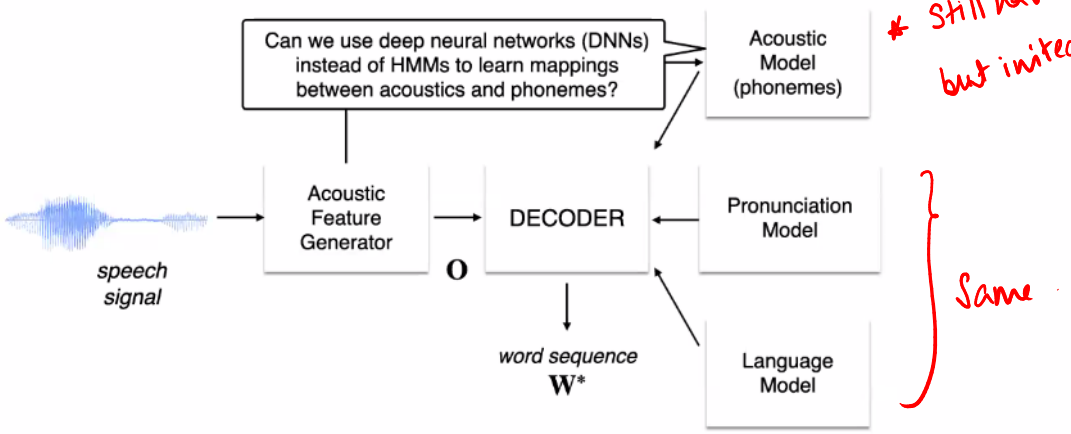
Decoder: Search Graph



Decoder: Search Graph



Hybrid HMM/DNN ASR System



* Still have HMM's but instead of GMM we have DNN

Comparison DNN-HMMs and GMM-HMMs

TASK	HOURS OF TRAINING DATA	DNN-HMM	GMM-HMM WITH SAME DATA	GMM-HMM WITH MORE DATA
SWITCHBOARD (TEST SET 1)	309	18.5	27.4	18.6 (2,000 H)
SWITCHBOARD (TEST SET 2)	309	16.1	23.6	17.1 (2,000 H)
ENGLISH BROADCAST NEWS	50	17.5	18.8	
BING VOICE SEARCH (SENTENCE ERROR RATES)	24	30.4	36.2	
GOOGLE VOICE INPUT	5,870	12.3		16.0 (>> 5,870 H)
YOUTUBE	1,400	47.6	52.3	

Hybrid DNN-HMM systems consistently outperform GMM-HMM systems (sometimes even when the latter is trained with lots more data)

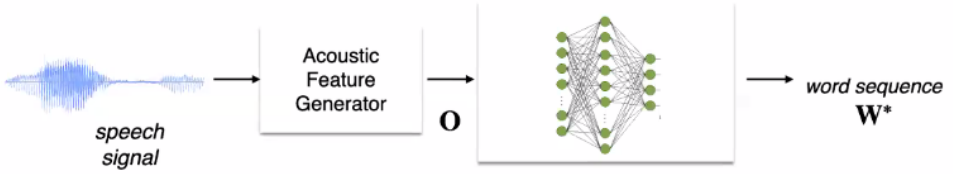
Table from G. Hinton, et al., "Deep Neural Networks for Acoustic Modeling in Speech Recognition", IEEE Signal Processing Magazine, 2012

Limitations of Cascaded ASR Systems

- Frame-level training targets derived from HMM-based alignments
- Pronunciation dictionaries are used to map from words to phonemes; expensive resource to create
- Complex training process, and difficult to globally optimise
- [Limitation not specific to cascaded systems per se]
  Objective function optimized in neural networks very different from final evaluation metric (i.e. word transcription accuracy)

Cascaded ASR ⇒ End-to-end ASR

$$\mathbf{W}^* = \arg\max_{W} \Pr(\mathbf{W} \mid \mathbf{O})$$



Single end-to-end model that directly learns a mapping from speech to text

End-to-End ASR Systems

- All components trained jointly as a single end-to-end model
- Trained using pairs of speech clips and their corresponding text transcripts
- End-to-end models, with sufficient data, sometimes outperform conventional ASR systems [1,2]

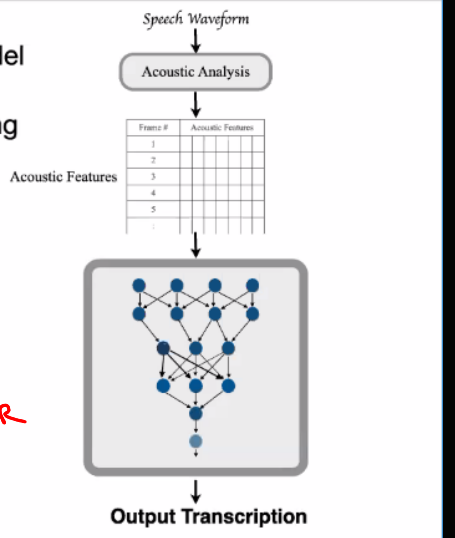
	dev	test
DNN-HMM	4.0	4.4
E2E (Attention)	4.7	4.8

Librispeech-960

	dev	test
DNN-HMM	5.0	5.8
E2E (Attention)	14.7	14.7

Librispeech-100

ER



[1] "State-of-the-art speech recognition with sequence-to-sequence models", Chiu et al., 2018
[2] "RWTH ASR Systems for LibriSpeech: Hybrid vs Attention", Lüscher, Christoph, et al. , 2019

CTC (2016)

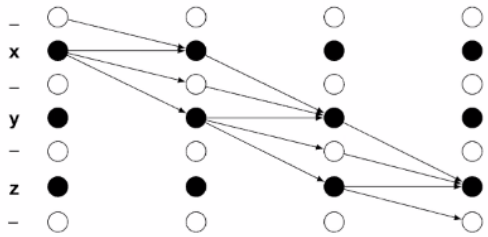
Connectionist Temporal Classification (CTC)

- Hybrid DNN-HMM systems typically require alignments between the acoustics and the word sequence during training telling you which label (e.g. phone or character) should be output at each speech frame
- Connectionist Temporal Classification (CTC) [1] is an objective function that considers all possible alignments and allows acoustic model training without requiring frame-level alignments

[1] Towards End-to-End Speech Recognition with Recurrent Neural Networks, Alex Graves and Navdeep Jaitly, ICML 2014

CTC: Prerequisites

- Augment the output vocabulary with an additional "blank" (_) label
- For a given label sequence, there can be multiple alignments: (x, y, z) could correspond to (x, _, y, _, _, z) or (_, x, x, _, y, z)
- Define a 2-step operator B that reduces a label sequence by: first, removing repeating labels and second, removing blanks. B("x, _, y, _, _, z") = B("_, x, x, _, y, z") = "x, y, z"



CTC: Overview

- CTC objective function is the probability of an output label sequence y given an utterance x (by summing over all possible alignments for y provided by $B^{-1}(y)$):

$$\mathrm{CTC}(x, y) = \Pr(y \mid x) = \sum_{a \in B^{-1}(y)} \Pr(a \mid x)$$

$$= \sum_{a \in B^{-1}(y)} \prod_{t=1}^{T} \Pr(a_t \mid x)$$

- Efficient forward+backward algorithm to compute this loss function and its gradients [GJ14]

Image from: https://distill.pub/2017/ctc/

Sequence-to-sequence Models

- CTC makes an assumption that the network outputs at different time steps are conditionally independent given the inputs
- The Listen, Attend and Spell [LAS] network [1] makes *no independence assumptions* about the probability distribution of the output sequences given the input

$$P(\mathbf{y}|\mathbf{x}) = \prod_i P(y_i|\mathbf{x}, y_{<i})$$

[1]: Chan et al., Listen, Attend and Spell: A Neural Network for LVCSR, ICASSP 2016

Sequence-to-sequence Models

- CTC makes an assumption that the network outputs at different time steps are conditionally independent given the inputs
- The Listen, Attend and Spell [LAS] network [1] makes *no independence assumptions* about the probability distribution of the output sequences given the input

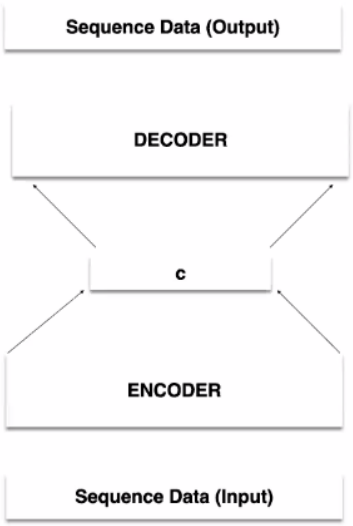
$$P(\mathbf{y}|\mathbf{x}) = \prod_i P(y_i|\mathbf{x}, y_{<i})$$

- Based on the sequence-to-sequence with attention framework

[1]: Chan et al., Listen, Attend and Spell: A Neural Network for LVCSR, ICASSP 2016

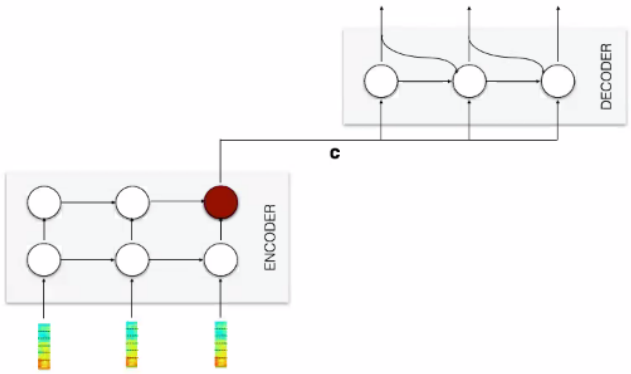
Sequence to sequence Models

Encoder-decoder architecture



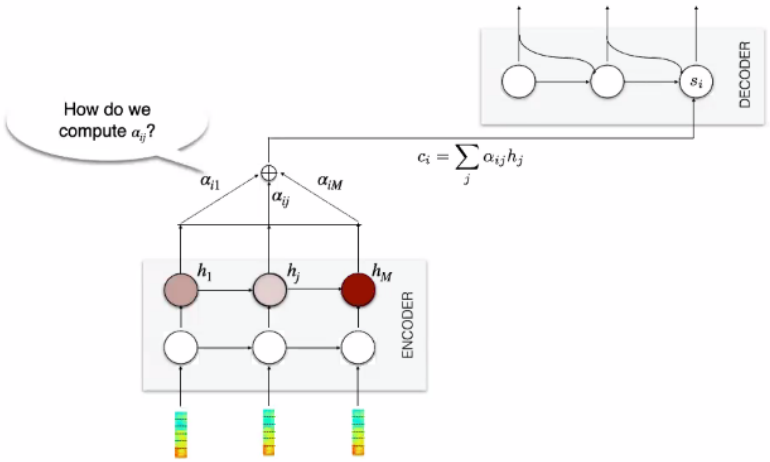
Sequence-to-sequence Models

Encoder-decoder architecture



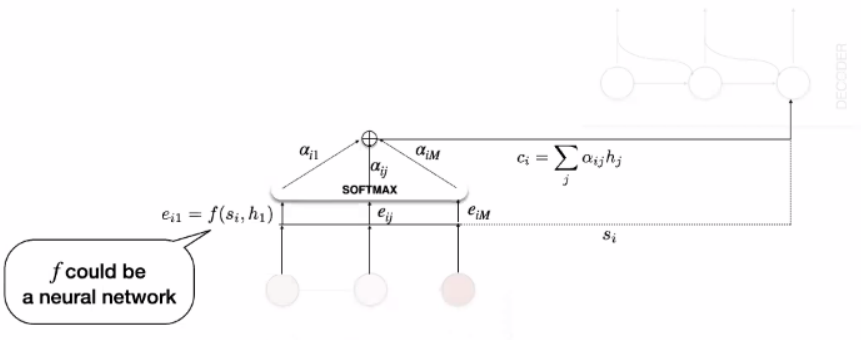
Sequence-to-sequence Models

Encoder-decoder architecture

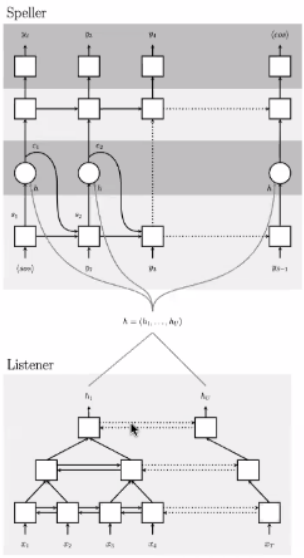


Sequence-to-sequence Models

Encoder-decoder architecture



The LAS Model



- The Listen, Attend & Spell (LAS) architecture is a sequence-to-sequence model consisting of
  - a Listener (Listen): An acoustic model encoder. Deep BLSTMs with a pyramidal structure: reduces the time resolution by a factor of 2 in each layer.
  - a Speller (AttendAndSpell): An attention-based decoder. Consumes $\mathbf{h}$ and produces a probability distribution over characters.

$$\mathbf{h} = \text{Listen}(\mathbf{x})$$
$$P(y_i|\mathbf{x}, y_{<i}) = \text{AttendAndSpell}(y_{<i}, \mathbf{h})$$

Image from: Chan et al., Listen, Attend and Spell: A NN for LVCSR, ICASSP 2016

Attend and Spell

- Produces a distribution over characters conditioned on all characters seen previously

$$c_i = \text{AttentionContext}(s_i, \mathbf{h})$$
$$s_i = \text{RNN}(s_{i-1}, y_{i-1}, c_{i-1})$$
$$P(y_i|\mathbf{x}, y_{<i}) = \text{CharacterDistribution}(s_i, c_i)$$

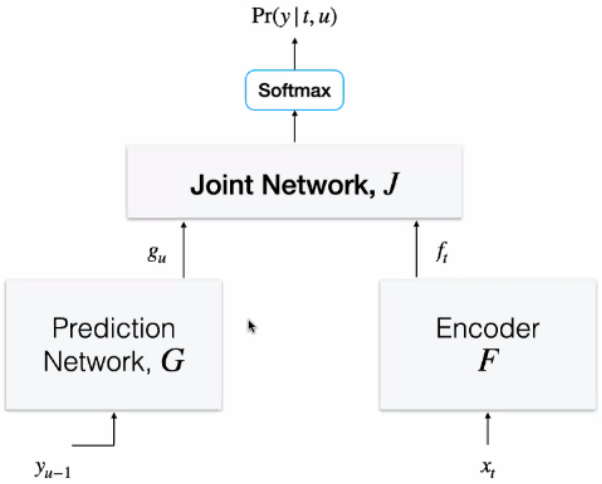
- At each decoder time-step *i*, AttentionContext computes a score for each encoder step *u*, which is then converted into softmax probabilities that are linearly combined to compute c_i

$$e_{i,u} = \langle \phi(s_i), \psi(h_u) \rangle$$
$$\alpha_{i,u} = \frac{\exp(e_{i,u})}{\sum_{u'} \exp(e_{i,u'})}$$
$$c_i = \sum_u \alpha_{i,u} h_u$$

Limitations of CTC / Attention-based Models

- CTC makes a frame independence assumption i.e., outputs are assumed to be conditionally independent of each other
- For CTC to work, the length of the output sequence has to necessarily be smaller than the length of the input sequence
- Attention-based models do not take advantage of the fact that the alignment between the speech and transcription is *monotonic*

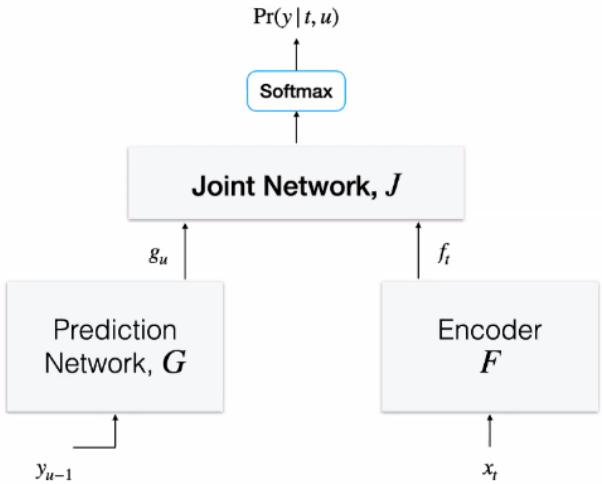
RNN Transducer [1]



- Encoder network: $f_t = F(x_t)$
- Prediction network: $g_u = G(y_{u-1})$
- Joint network: $h_{t,u} = J(f_t, g_u)$ $\Pr(y \mid t, u) = \text{softmax}(h_{t,u})$

[1] Alex Graves, "Sequence Transduction with Recurrent Neural Networks", 2012

RNN Transducer [1]



- During decoding:
  - Read an x_t
  - Produce output labels until an empty label is encountered
  - If an empty label is produced as output, read the next input x_{t+1}
  - Repeat until the entire input sequence is read

Other Considerations with End-to-End Models

- What units should be used in the output vocabulary? [Rao et al. 2017, Chiu et al. 2018]
- How to improve the training of end-to-end models? Scheduled sampling, label smoothing [Bengio et al. 2015], etc.
- Use of an external language model? [Kannan et al. 2017]

	Gujarati	Telugu
Hybrid	43.2	46.8
Hybrid + RNNLM	34.0	40.0

[1] Rao et al. 2017, "Exploring architectures, data and units for streaming end-to-end speech recognition with RNN-transducers", ASRU 2017
[2] Chiu et al. 2018, "State-of-the-art Speech Recognition With Sequence-to-Sequence Models", ICASSP 2018
[3] Bengio et al. 2015, "Scheduled Sampling for Sequence Prediction with Recurrent Neural Networks", NeurIPS 2015
[4] Kannan et al. 2017, "An analysis of incorporating an external language model into a sequence-to-sequence model", ICASSP 2018

wav2vec

- Algorithm that uses raw audio to learn speech representations ("**self-supervised**" approach)
  - Encoder network embeds raw audio into a latent representation ($f : \mathcal{X} \rightarrow \mathcal{Z}$) and a context network combines multiple encoded representations into a contextualised embedding ($g : \mathcal{Z} \rightarrow \mathcal{C}$)

wav2vec

- Algorithm that uses raw audio to learn speech representations ("**self-supervised**" approach)
  - Encoder network embeds raw audio into a latent representation ($f : \mathcal{X} \rightarrow \mathcal{Z}$) and a context network combines multiple encoded representations into a contextualised embedding ($g : \mathcal{Z} \rightarrow \mathcal{C}$)
  - Train the model to minimize the following contrastive loss:

$$\mathcal{L}_k = -\sum_{i=1}^{T-k} \log \frac{\exp(\text{sim}(c_i, z_{i+k}))}{\sum_{\tilde{z}} \exp(\text{sim}(c_i, \tilde{z}))}$$

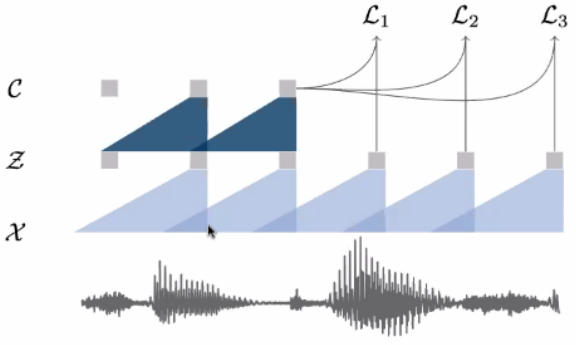
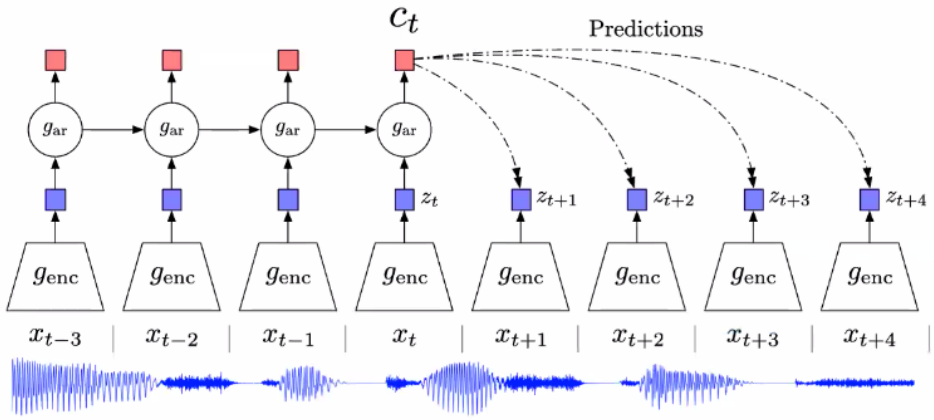


Image from: https://arxiv.org/pdf/1904.05862.pdf

Contrastive Predictive Coding



$$\mathcal{L}_k = -\sum_{i=1}^{T-k} \log \frac{\exp(\text{sim}(c_i, z_{i+k}))}{\sum_{\tilde{z}} \exp(\text{sim}(c_i, \tilde{z}))}$$

Contrastive Loss

Image from: https://arxiv.org/abs/1807.03748

wav2vec

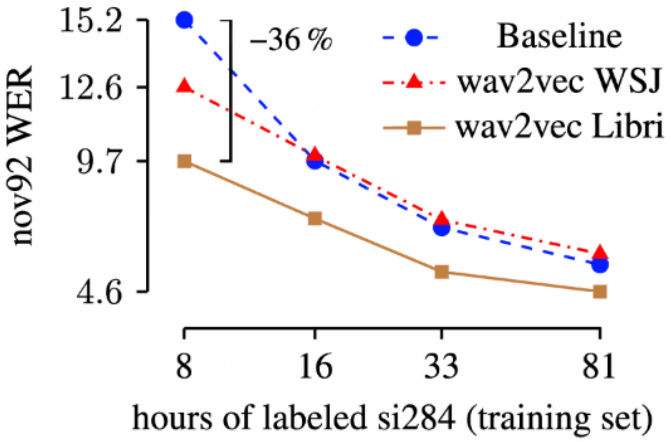


Image from: https://arxiv.org/pdf/1904.05862.pdf

wav2vec 2.0: Learning Speech Representations from Raw Audio

- Similar to wav2vec. Outputs from the encoder are further quantized.
- Masks spans of speech representations (as in masked language modelling for BERT [1])
- Training objective is to recover the masked representations among a set of distractors.

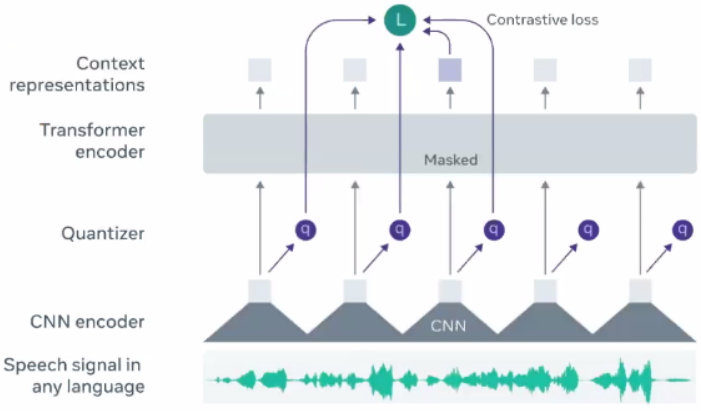


Image from:https://ai.facebook.com/blog/wav2vec-20-learning-the-structure-of-speech-from-raw-audio/
[1]: https://arxiv.org/abs/1810.04805

wav2vec 2.0: Results on English

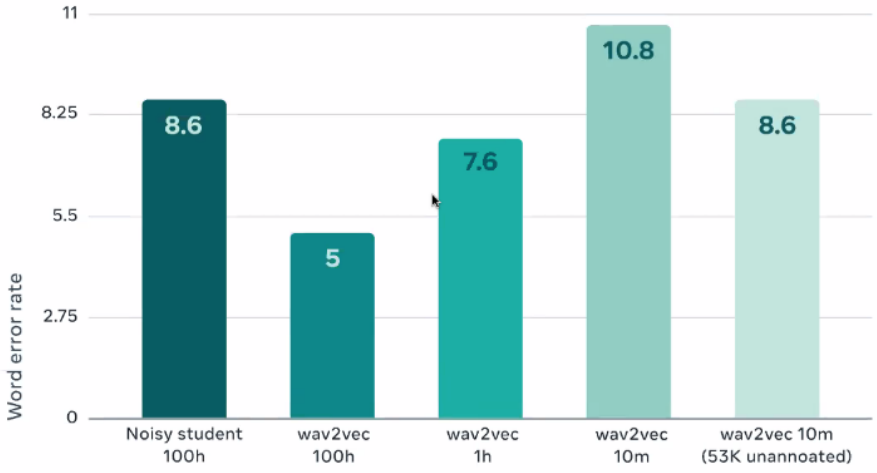


Image from:https://ai.facebook.com/blog/wav2vec-20-learning-the-structure-of-speech-from-raw-audio/

XLSR: Multilingual Self-supervised Speech Model

- wav2vec 2.0 model trained on speech in 53 languages
- Training objective is to recover the masked representations within a set of distractors
- Cross-lingual pretraining significantly outperforms monolingual pretraining

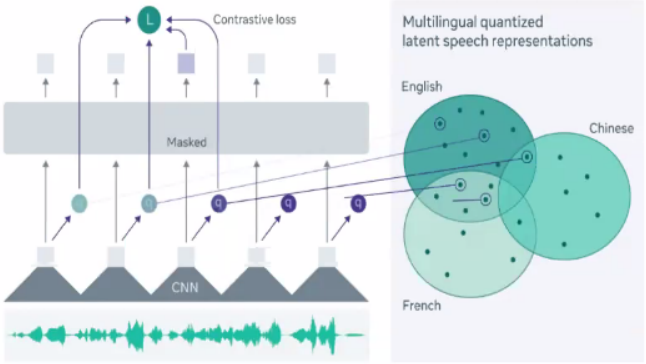


Image from:https://ai.facebook.com/blog/wav2vec-20-learning-the-structure-of-speech-from-raw-audio/
[1]: https://arxiv.org/abs/1810.04805

XLSR Results

Model	D	#pt	#ft	es	fr	it	ky	nl	ru	sv	tr	tt	zh	Avg
Number of pretraining hours per language				168h	353h	90h	17h	29h	55h	3h	11h	17h	50h	793h
Number of fine-tuning hours per language				1h	1h	1h	1h	1h	1h	1h	1h	1h	1h	10h
Baselines from previous work														
m-CPC† (Rivière et al., 2020)	LS_{100h}	10	1	38.7	49.3	42.1	40.7	44.4	45.2	48.8	49.7	44.0	55.5	45.8
m-CPC† (Rivière et al., 2020)	LS_{360h}	10	1	38.0	47.1	40.5	41.2	42.5	43.7	47.5	47.3	42.0	55.0	44.5
Fer et al.† (Fer et al., 2017)	BBL_{all}	10	1	36.6	48.3	39.0	38.7	47.9	45.2	52.6	43.4	42.5	54.3	44.9
Our monolingual models														
XLSR-English	CV_{en}	1	1	13.7	20.0	19.1	13.2	19.4	18.6	21.1	15.5	11.5	27.1	17.9
XLSR-Monolingual	CV_{mo}	1	1	6.8	10.4	10.9	29.6	37.4	11.6	63.6	44.0	21.4	31.4	26.7
Our multilingual models														
XLSR-10 (unbalanced)	CV_{all}	10	1	9.7	13.6	15.2	11.1	18.1	13.7	21.4	14.2	9.7	25.8	15.3
XLSR-10	CV_{all}	10	1	9.4	14.2	14.1	8.4	16.1	11.0	20.7	11.2	7.6	24.0	13.6
XLSR-10 (separate vocab)	CV_{all}	10	10	10.0	13.8	14.0	8.8	16.5	11.6	21.4	12.0	8.7	24.5	14.1
XLSR-10 (shared vocab)	CV_{all}	10	10	9.4	13.4	13.8	8.6	16.3	11.2	21.0	11.7	8.3	24.5	13.8
Our multilingual models (Large)														
XLSR-10	CV_{all}	10	1	7.9	12.6	11.7	7.0	14.0	9.3	20.6	9.7	7.2	22.8	12.3
XLSR-10 (separate vocab)	CV_{all}	10	10	8.1	12.1	11.9	7.1	13.9	9.8	21.0	10.4	7.6	22.3	12.4
XLSR-10 (shared vocab)	CV_{all}	10	10	7.7	12.2	11.6	7.0	13.8	9.3	20.8	10.1	7.3	22.3	12.2
Our Large XLSR-53 model pretrained on 56k hours														
XLSR-53	D_{53}	53	1	**2.9**	**5.0**	**5.7**	**6.1**	**5.8**	**8.1**	**12.2**	**7.1**	**5.1**	**18.3**	**7.6**

Image from: https://arxiv.org/pdf/2006.13979.pdf

Fine-Tune XLSR-Wav2Vec2 for low-resource ASR with 🤗 Transformers

Published March 12, 2021.

Update on GitHub

patrickvonplaten
Patrick von Platen

Open in Colab

Wav2Vec2 is a pretrained model for Automatic Speech Recognition (ASR) and was released in September 2020 by Alexei Baevski, Michael Auli, and Alex Conneau. Soon after the superior performance of Wav2Vec2 was demonstrated on the English ASR dataset LibriSpeech, *Facebook AI* presented XLSR-Wav2Vec2 (click here). XLSR stands for *cross-lingual speech representations* and refers to XLSR-Wav2Vec2's ability to learn speech representations that are useful across multiple languages.

https://huggingface.co/blog/fine-tune-xlsr-wav2vec2

Many Frameworks Available to Learn Speech Representations

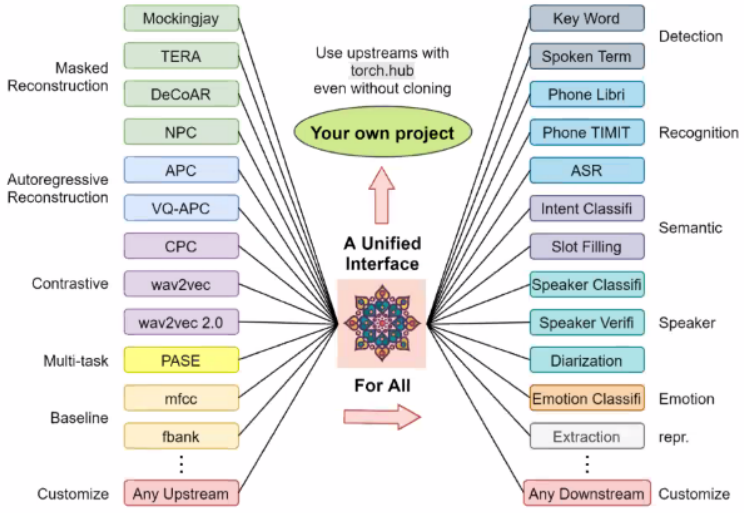
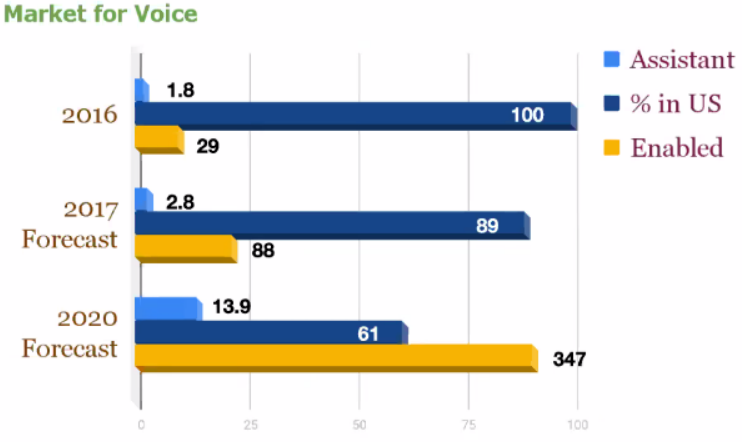


Image from: https://github.com/s3prl/s3prl

Exciting Time to do Speech Research



Exciting Time to do Speech Research

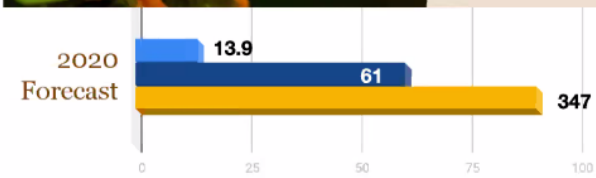
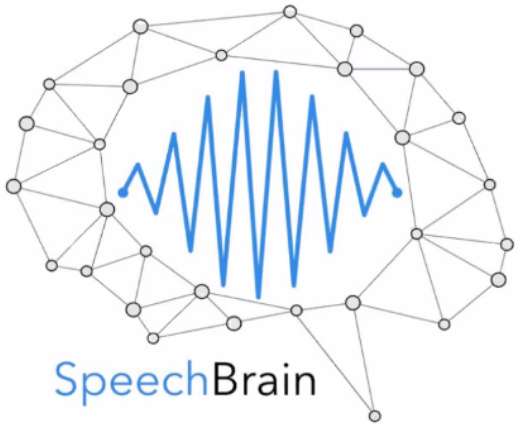


Image from: https://coqui.ai/

Exciting Time to do Speech Research



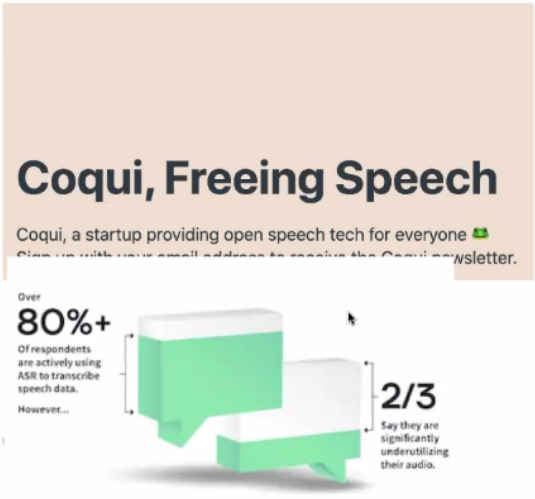


Image from: https://coqui.ai/
https://speechbrain.github.io/

What's next?

Need to do more…

- Robust to variations in age, accent and ability
- Handling noisy real-life settings with many speakers (e.g., meetings, parties)
- Handling downstream tasks that work with noisy ASR outputs
- Handling new languages/ dialects

What's next?

Need to do more…

- Robust to variations in age, accent and ability
- Handling noisy real-life settings with many speakers (e.g., meetings, parties)
- Handling pronunciation variability
- Handling new languages/ dialects

… with less

- Fast (real-time) decoding using limited computational power/ memory
- Faster training algorithms
- Reduce duplicated effort across domains/languages
- Reduce dependence on language-specific resources
- Train with less labeled data

Challenges (I)

- "Voice is the next big platform, unless you have an accent" [1]:
  - Non-native accents still pose a significant challenge to state-of-the-art ASR systems

[1] https://www.wired.com/2017/03/voice-is-the-next-big-platform-unless-you-have-an-accent/
Image from https://fairspeech.stanford.edu/

Challenges (I)

- "Voice is the next big platform, unless you have an accent" [1]:
  - Non-native accents still pose a significant challenge to state-of-the-art ASR systems


```
Target:  BUT YOU ARE JOKING
         BUT YOU ARE JOKING
         BUT YOU ARE JOKING
         BUT HERE CHOKING
```

- WERs range from 11% to 53% using a state-of-the-art ASR system across eight different Indian accented English speech test sets [2]

[1] https://www.wired.com/2017/03/voice-is-the-next-big-platform-unless-you-have-an-accent/
Image from https://fairspeech.stanford.edu/
Audio utterances from Mozilla CommonVoice Speech Corpus
[2] Awasthi et al., ICASSP 2021, https://arxiv.org/abs/2103.03142

Challenges (II)

- Code switching is widely used in multilingual countries like India, and pose interesting challenges for computational models
  - Examples of code switching [1]:
    "I was going for a movie yesterday. Raaste men mujhe Sudha mil gayi"
    "Main kal movie dekhne jaa rahi thi aur raaste me I met Sudha"
- Large diversity in how code-switching can manifest, even within a single sentence [2]

पर हंसी चिकित्सा ने मेरा जीवन बदल दिया वास्तव में

But laughter therapy ने मेरी life बदल दी actually

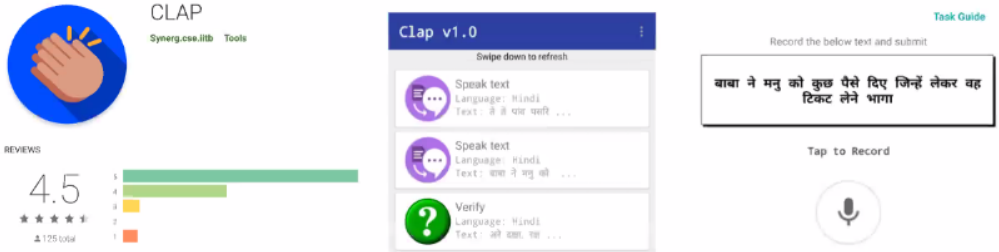
But laughter therapy ने really में मेरी life change कर दी

पर हंसी therapy ने मेरी life बदल दिया वास्तव में

[1] Example from "I am borrowing ya mixing? An analysis of English-Hindi code mixing in FB", Bali et al., ACL workshop, 2014
[2] Example from "From Machine Translation to Code Switching: Generating High-quality Code-switched Text", Tarunesh, Kumar, Jyothi, ACL 2021

Crowdsourcing Speech in Indian Languages

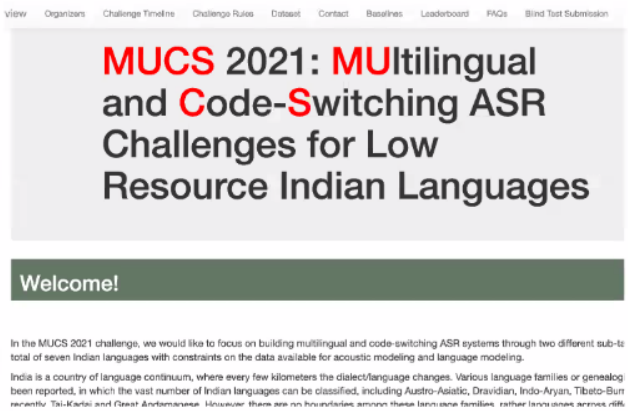
- Crowdsourcing for Language Processing (**CLAP**): Android app on Google Playstore for crowdsourcing labeled speech in Indian languages [1]
  - Target: By October 2021, release well-curated speech corpora in 4-6 Indian languages (from speakers all over India)



[1] https://www.cse.iitb.ac.in/clap/

MUCS 2021 Workshop

Multilingual + Code-switching ASR, https://navana-tech.github.io/MUCS2021/



#	Team Name	Hindi-English (% WER)	Bengali-English (% WER)	Average (% WER)
1	CSTR	15.64	22.61	19.12
2	JHU-CLSP/GoVivace	15.51	24.67	20.09
3	Sayint	18.78	24.34	21.56
4	KARI	20.49	24.07	22.28
5	Bytedance-SA	19.65	25.29	22.47
6	IITM-SMT-Lab	20.97	26.69	23.83
7	Ekstep	20.75	26.96	23.85
8	TUTU	22.3	28.04	25.17
9	MCSASR	22.54	28.57	25.55
10	Jio Speech	23.83	30.15	26.99
11	INDIGO-IITG	23.78	31.2	27.49
12	Baseline	23.8	31.7	27.75