Introduction to Automatic Speech Recognition

P. R. Sullivan

UBC MDS-CL, COLX 585, Spring 2021

Table of Contents

Objective

This is a high level talk focusing on a birds-eye view of research in Speech Technologies, looking at trends in research (including history and current models) and challenges compared with text-based approaches.
Hands on tutorials with a particular tool (Facebook's Wav2Vec2 implemented in Huggingface's Transformer library) coming next week.

Table of Contents

Why bother with speech tasks?

Lots of challenges, however it represents a major area of opportunity, with lots of low-hanging fruit.

- Rise of multimodality: With TikTok, Instagram, YouTube etc. tasks that used to be focused on text, need to take into account audio/video.
- Speech-only tasks: Subtitling, Translation of non-written languages, Simultaneous in-person translation.
- Linguistics: Prosody and other linguistic cues carry lots of information not present in text, potentially making speech based-models powerful HCI tools.

Challenges of working with Speech

That said...

- Data constraints (Size of vocab, #hrs, languages available, domain)
- Linguistic variation (non-native speech, dialect, disfluency etc)
- Segmentation! What does a 'period' sound like?
- Segment Length. Sentences containing 10 tokens might take up to 1000 "frames" of audio.

Metrics and Data Preparation

- ASR uses Word Error Rate (WER), Character Error Rate (CER), and in the case of SLT we use BLEU (not without reservation), segment length, and Translation Error Rate (TER).
- Models can be built to take raw audio (e.g. wav, sphere, mp3 etc.), Mel Frequency Cepstral Coefficient (MFCC), or log-Mel Spectrograms.
- Audio usually recorded at 16khz or 8khz sample rate (don't mix!)

Types of Speech Tasks

- Automatic Speech Recognition
- Sound event detection
  Speaker ID
- Spoken Language Translation (aka Speech Translation)

Table of Contents

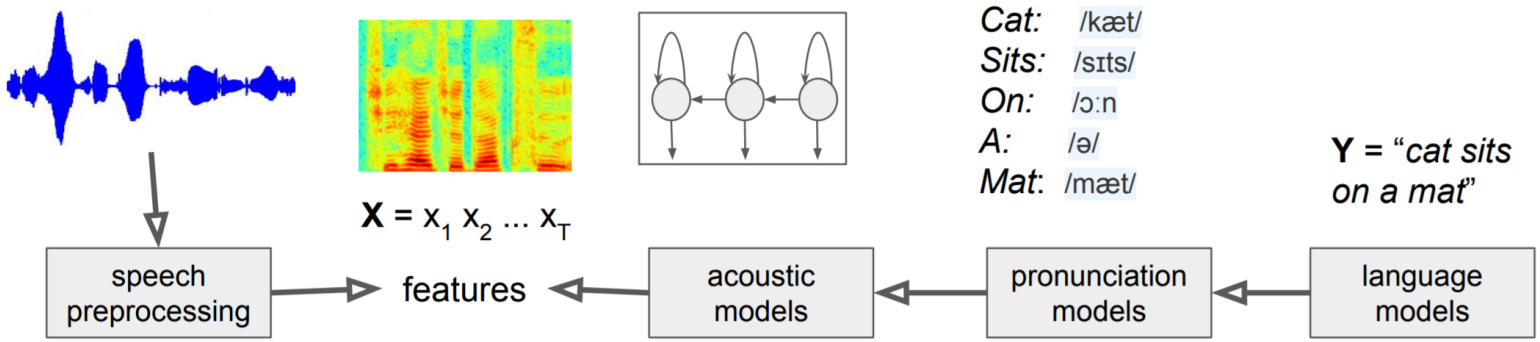
Common ASR Models

- Traditionally: Hidden Markov Model based approaches. HMM+DNN or HMM+GMM. (Work well, but one major downside).
- End-to-End approaches: CTC, ASG, Seq2seq with attention (or a hybrid). Recently: Transformer-based models!

HMM/GMM (aka "Traditional ASR")

Hidden Markov Models used to model sequence of emissions (given by a Gaussian Mixture Model). Good performance, regularly SOTA until 2019

- Parts trained separately (tools like Kaldi manage this)
- Hand-tuned parameters
- Need TIMIT-style datasets (error prone and tedious)



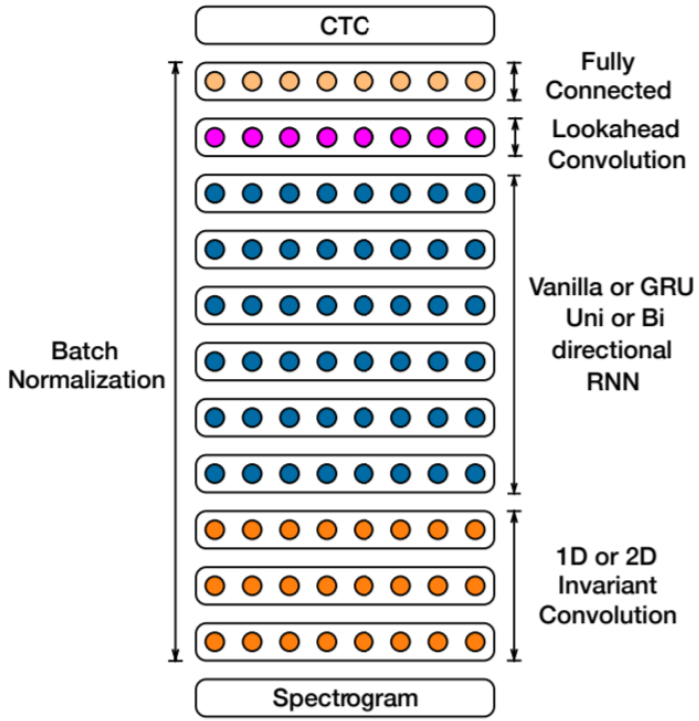
$$\mathbf{Y}^* = \arg\max_{\mathbf{Y}} p(\mathbf{X}|\mathbf{Y})\, p(\mathbf{Y})$$

[1] https://web.stanford.edu/class/archive/cs/cs224n/cs224n.1174/lectures/cs224n-2017-lecture12.pdf

CTC (e.g. Deep Speech 2)

- CTC loss allows automatic alignment with target text at the frame level (see next slide)
- Best used with decoding strategy (Beam) and a Language Model

Deep Speech 2 [1]



CTC Cont.

CTC Algorithm

- Input Sequence > Target Sequence
- Special "blank" character needed (shown as ϵ) with PyTorch this is always index 0.
- Can't use vanilla beam search due to blank symbol.

We start with an input sequence, like a spectrogram of audio.

The input is fed into an RNN, for example.

The network gives $p_t(a \mid X)$, a distribution over the outputs {h, e, l, o, ϵ} for each input step.

With the per time-step output distribution, we compute the probability of different sequences

By marginalizing over alignments, we get a distribution over outputs.

[1] from `https://distill.pub/2017/ctc/`

Select CTC Papers

- [2] Towards End-to-End Speech Recognition with Recurrent Neural Networks — The first true E2E system, RNN with CTC loss. Novel modification of Beam search to work with CTC and Language Model (although in practice difficult to get working well).
- [3] First-Pass Large Vocabulary Continuous Speech Recognition using Bi-Directional Recurrent DNNs — This paper introduces a better method for Beam search decoding of CTC output (prefix beam search)
- [1] Deep Speech 2 — Baidu refined their original purely RNN approach (Deep Speech) with the inclusion of CNN layers and batch normalization.

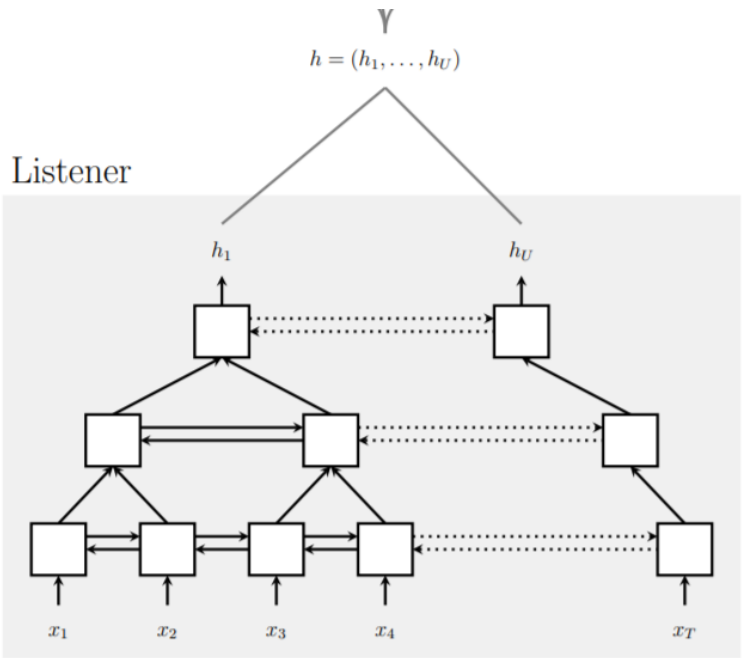
Seq2seq (e.g. Listen Attend and Spell)

- No independence assumption made or frame-level prediction
- Seq2Seq models sometimes end early on long strings
- Some variety in design, but mainly RNN-based Encoder and Decoder with attention.

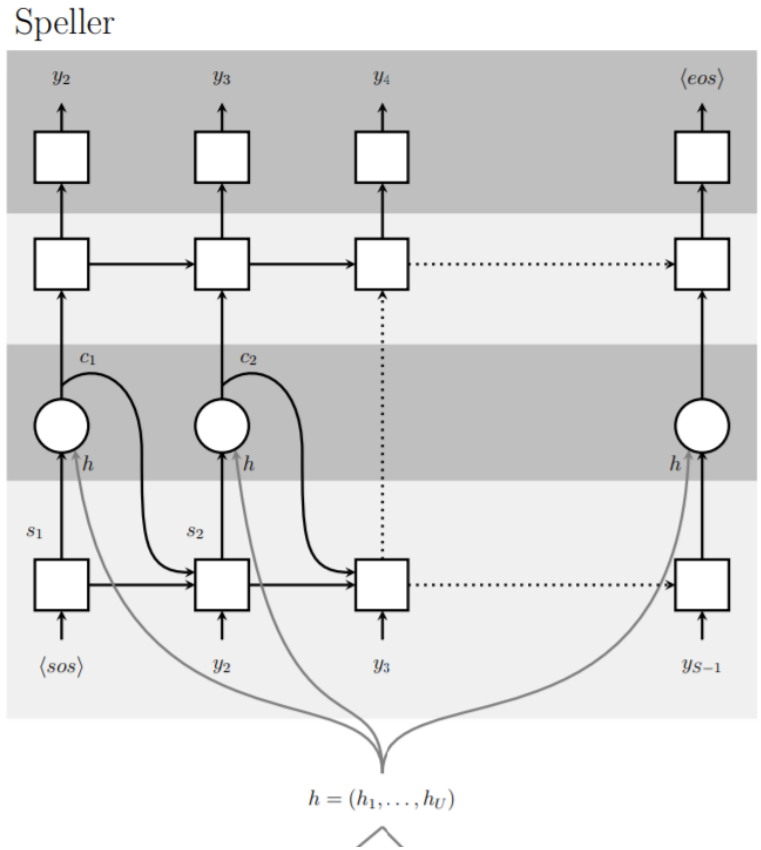
1

LAS [4]

Listener (pBLSTM)

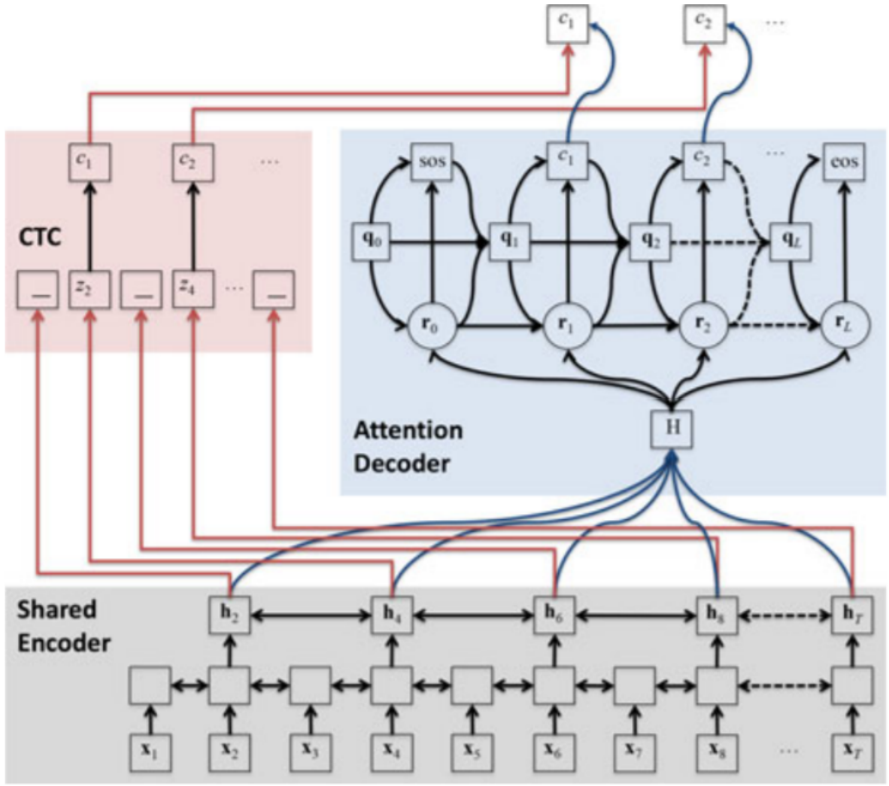


Speller (LSTM-Transducer)



Hybrid CTC/Attention

CTC independent of Seq2Seq prediction, thus can improve performance and alignment, including fixing early stopping.



[1] Image from [5]

Select Seq2Seq Papers

- [4] Listen, Attend, and Spell — most influential of the seq2seq models, when combined with data augmented through SpecAugment [6] gives SOTA performance due to being able to massively increase model size.
- [7] Streaming End-to-end Speech Recognition For Mobile Devices — RNN-Transducers can be used with CTC, allowing for a seq2seq model that outputs continuous predictions. This shows how you can build lightweight models for real-time tasks. Notably avoids using attention.
- [8] Improved training of end-to-end attention models for speech recognition — Standard LSTM Encoder-Decoder model, using supplemental CTC loss to aid convergence (not in decoding). Use of BPE significantly improves performance.

Adapting Transformer to Speech Tasks

Straightforward: Just replace Hybrid CTC/Attention Encoder and Decoder with Transformer-Encoder/Decoder. But some drawbacks:

- Transformer grows as $O(L^2 d)$ vs $O(L d^2)$ for RNN
- Positional Encoding hinders performance

Conformer [9]

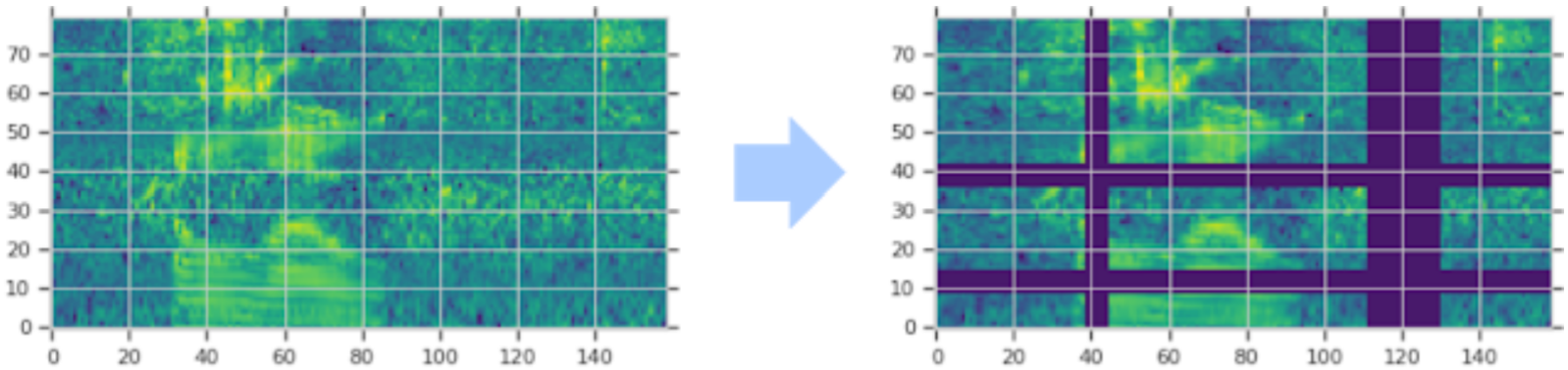
One example of a newer Transformer-like model aimed for ASR is the Conformer model.
A new SOTA model modifies the traditional Transformer architecture by adding a convolutional module and Transformer-XL style relative position multihead attention. The benefits of this approach include extremely competitive performance on Librispeech even with extremely small model size (10M parameters compares favourably with the best Convnet-based approach, and larger models outperform LAS and other transformer based systems), even without a language model.

Select Transformer Papers for ASR

- [10]A comparative study on transformer vs RNN in speech applications — A comparison of Transformers vs. RNNs for not only ASR, but also ST and TTS. Gives concrete suggestions on training and using transformer.

SpecAugment [6]

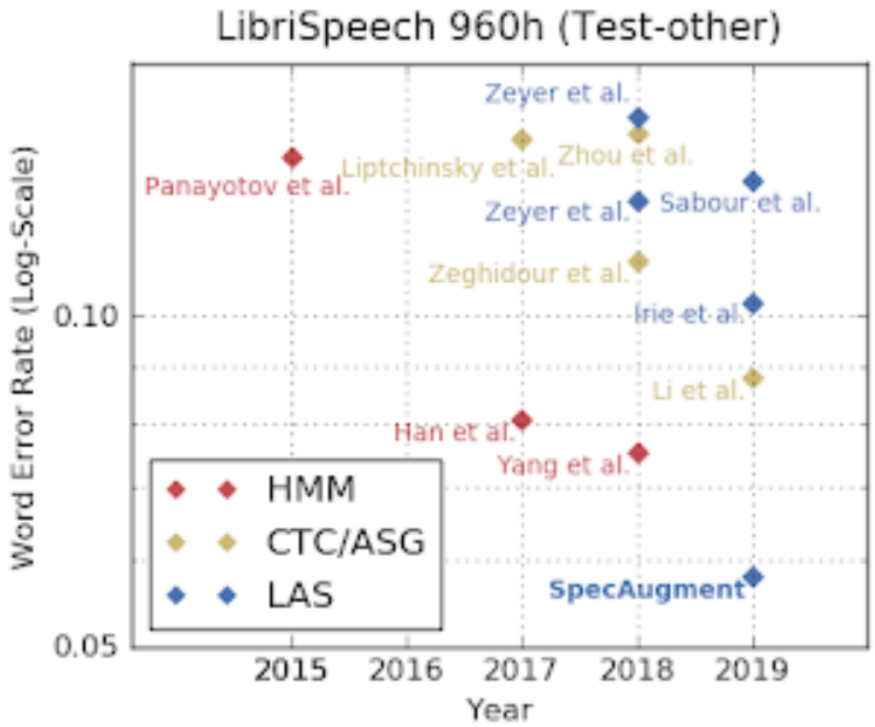
Data Augmentation makes models significantly more robust, allowing you to significantly increase size without drawback. SpecAugment does this by masking (time, channel), and warping across time.



[1] Image from `https://ai.googleblog.com/2019/04/specaugment-new-data-augmentation.html`

State of the Art circa 2019

Finally improvement past HMM models! Of interest the Li et al. model is a purely convolutional model, and Irie et al. is one of the earliest succesful Transformer approaches.



[1] Image from https://ai.googleblog.com/2019/04/specaugment-new-data-augmentation.html

Summing Up (ASR)

- CTC allows for auto-alignment and first E2E systems
- Seq2Seq w/Attention get around per-frame prediction issues and markov assumptions.
- Transformers replacing RNNs in ASR, allowing for faster training and better accuracy, however, with some issues.
- Data augmentation is vital for success of models.

References I

[1] D. Amodei, S. Ananthanarayanan, R. Anubhai, J. Bai, E. Battenberg, C. Case, J. Casper, B. Catanzaro, Q. Cheng, G. Chen *et al.*, "Deep speech 2: End-to-end speech recognition in english and mandarin," in *International conference on machine learning*, 2016, pp. 173–182.

[2] A. Graves and N. Jaitly, "Towards end-to-end speech recognition with recurrent neural networks," in *International conference on machine learning*, 2014, pp. 1764–1772.

[3] A. Y. Hannun, A. L. Maas, D. Jurafsky, and A. Y. Ng, "First-pass large vocabulary continuous speech recognition using bi-directional recurrent dnns," *arXiv preprint arXiv:1408.2873*, 2014.

References II

[4] W. Chan, N. Jaitly, Q. Le, and O. Vinyals, "Listen, attend and spell: A neural network for large vocabulary conversational speech recognition," in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2016, pp. 4960–4964.

[5] S. Watanabe, T. Hori, S. Kim, J. R. Hershey, and T. Hayashi, "Hybrid ctc/attention architecture for end-to-end speech recognition," *IEEE Journal of Selected Topics in Signal Processing*, vol. 11, no. 8, pp. 1240–1253, 2017.

[6] D. S. Park, W. Chan, Y. Zhang, C.-C. Chiu, B. Zoph, E. D. Cubuk, and Q. V. Le, "Specaugment: A simple data augmentation method for automatic speech recognition," *arXiv preprint arXiv:1904.08779*, 2019.

References III

[7] Y. He, T. N. Sainath, R. Prabhavalkar, I. McGraw, R. Alvarez, D. Zhao, D. Rybach, A. Kannan, Y. Wu, R. Pang *et al.*, "Streaming end-to-end speech recognition for mobile devices," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 6381–6385.

[8] A. Zeyer, K. Irie, R. Schlüter, and H. Ney, "Improved training of end-to-end attention models for speech recognition," *arXiv preprint arXiv:1805.03294*, 2018.

[9] A. Gulati, J. Qin, C.-C. Chiu, N. Parmar, Y. Zhang, J. Yu, W. Han, S. Wang, Z. Zhang, Y. Wu *et al.*, "Conformer: Convolution-augmented transformer for speech recognition," *arXiv preprint arXiv:2005.08100*, 2020.

References IV

[10] S. Karita, N. Chen, T. Hayashi, T. Hori, H. Inaguma, Z. Jiang, M. Someki, N. E. Y. Soplin, R. Yamamoto, X. Wang *et al.*, "A comparative study on transformer vs rnn in speech applications," in *2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*. IEEE, 2019, pp. 449–456.