Introduction to Automatic Speech Recognition and Speech Translation

P. R. Sullivan[1]

[1]School of Information
UBC

UBC DL-NLP, July 2020

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 (ASR/SLT) aimed for later this summer.

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 (wav), Mel Frequency Cepstral Coefficient (MFCC), or log-Mel Spectrograms.
- For human speech we generally take 23 ms samples (and choose appropriate window, nfft for that).

Table of Contents

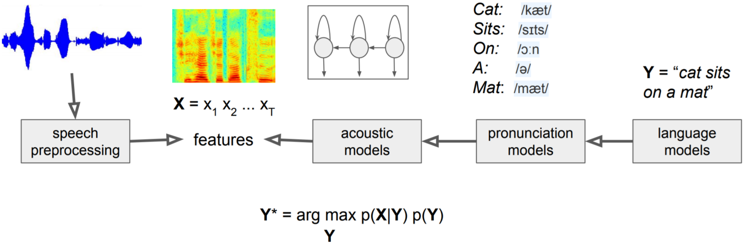
Common ASR Models

- 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. Recently: Transformer-based models (however, not without some hickups)!

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)

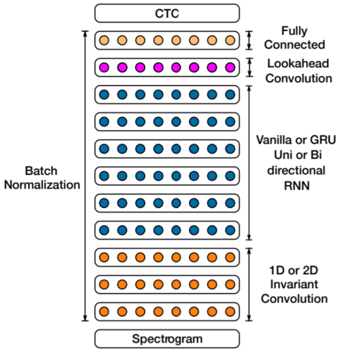


[1] https://web.stanford.edu/class/archive/cs/cs224n/cs224n.1174/

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

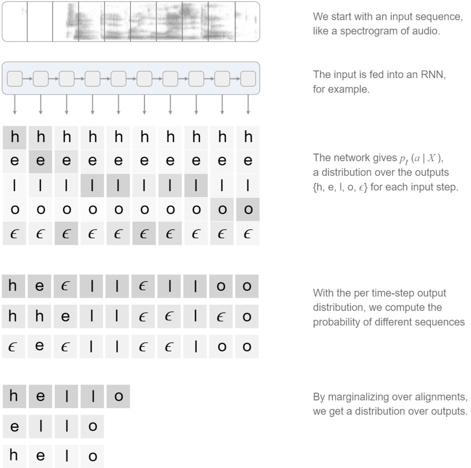
Deep Speech 2



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.



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

Select CTC Papers

- [1] 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).
- [2] 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)
- [3] Deep Speech 2 — Baidu refined their original purely RNN approach (Deep Speech) with the inclusion of CNN layers and batch normalization.

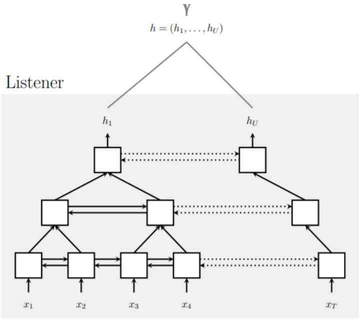
Seq2seq (e.g. Listen Attend and Spell)

- 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.

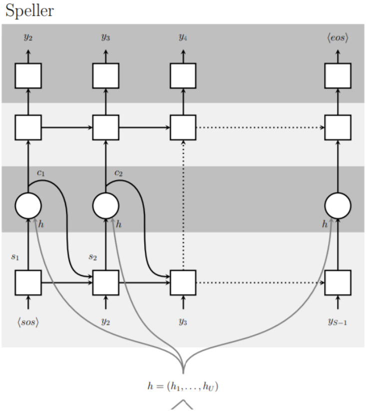
1

LAS [4]

Listener (pBLSTM)

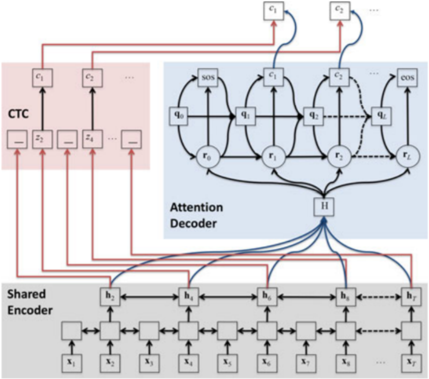


Speller (LSTM-Transducer)



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 [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:

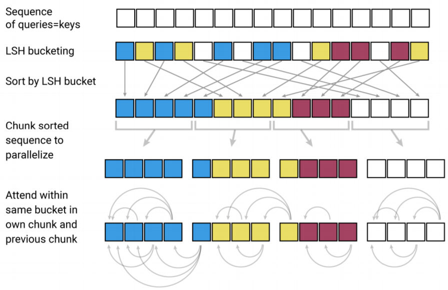
- Transformer grows as $O(L^2 d)$ vs $O(Ld^2)$ for RNN
- Positional Encoding hinders performance

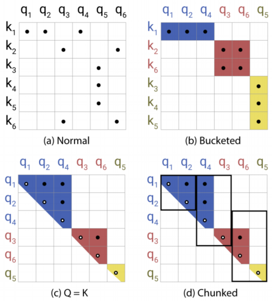
Reformer [ICLR 2020]

Reformer optimizes this by using locality sensitive hashing attention, as well as making memory efficiency adjustments (reversible layers, chunking during ff).

- Significantly faster than Transformer for longer sequence lengths $O(L\log(L))$
- Significantly smaller memory footprint (Can fit 20 layer model on 16gb GPU)
- Huggingface Transformer implementation (in progress)
- Negligible loss of accuracy
- However, hasn't been used for ASR/ST yet
- No pre-trained versions

Location Sensitive Hashing Attention cont.





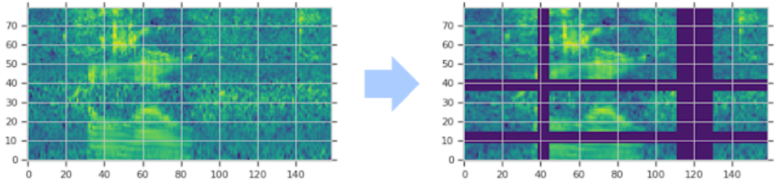
[1]from [9]

Select Transformer Papers for ASR

- [10] Language modeling with deep Transformers — An early effort to use Transformer, focuses on training deep models that can act as language models. They find the positional encoding produces results with worse perplexity, implying that deep Transformers are able to understand context without them.
- [11]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.
- [12] A simplified fully quantized transformer for end-to-end speech recognition — This paper focuses on parts of the transformer model that can be simplified. While full quantization may not be desired, their approach to positional encodings is useful.

SpecAugment

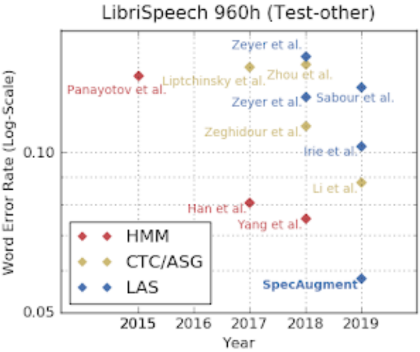
Data Augmentation makes models significantly more robust, allowing you to significantly increase size without drawback. SpecAugment does this by masking (time, channel), and warping across time.



[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.

Table of Contents

SLT

Because the dominant approach in SLT is still a cascade based approach (ASR + MT of output text), advances in ASR (and MT) carry over to SLT for the most part. As of IWSLT 2020, most models are Transformer based using modern training procedures (SpecAugment and additional pre-training). That said there are some noteable issues:

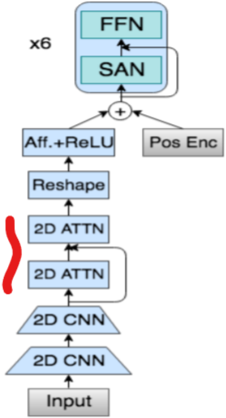
- Training becomes much trickier. Often need to pre-train Encoder on ASR set to ensure convergance.
- Segmentation of audio matters much more, as errors in segmentation can impact translation accuracy.
- For simultaneous SLT and subtitling, BLEU becomes a poor metric as it encourages over-generation. These tasks need implicit summarisization to work well.

S-Transformer

Standard model for IWSLT 2020

Key Change: Modify Transformer with 2D attention layers prior to Self-attention. (Results on MuST-C with log and Gaussian distance penalty for 2D attention)

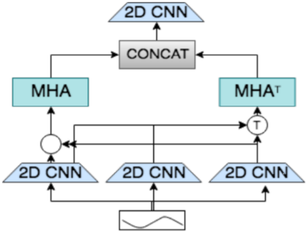
TGT	LSTM	log	Gauss	BIG+log	BIG+Gauss
De	12.9	**14.5**	14.4	**17.3**	16.2
Es	17.9	18.4	**18.6**	**20.8**	20.1
Fr	22.3	23.1	**24.0**	**26.9**	24.7
It	15.0	15.0	**15.6**	**16.8**	16.2
Nl	**18.2**	**18.1**	17.2	**18.8**	18.1
Pt	17.1	18.6	**19.7**	**20.1**	19.3
Ro	13.4	14.7	**15.0**	**16.5**	16.1
Ru	7.2	8.8	**9.1**	**10.5**	8.5



[1] Images From [13]

S-Transformer 2D attention

- Use 2D convs to create Q, K, V, matrices.
- Apply Multi-head attention on Q,K,V (as normal)
- concatenate and pass to final CNN



[1] Images From [13]

Summary IWSLT

- E2E models becoming (but not yet) competitive.
- Big hurtle is data, few large SLT corpora (MuST-C, CoVoST, Augmented-Librispeech), all of which have much smaller sizes than comporable ASR corpora (Librispeech 1000hr)

Table of Contents

ACL 2020 Papers

- Meta-Transfer Learning for Code-Switched Speech Recognition
- Curriculum Pre-training for End-to-End Speech Translation
- Curriculum Learning for Natural Language Understanding
- Phone Features Improve Speech Translation

MAML for Code-switched speech recognition

Use Meta-learning to harness large monolingual dataset through updating parameters on how it does at a task and then calculating the final loss based on how that update would do on the target dataset.

Algorithm 1 Meta-Transfer Learning

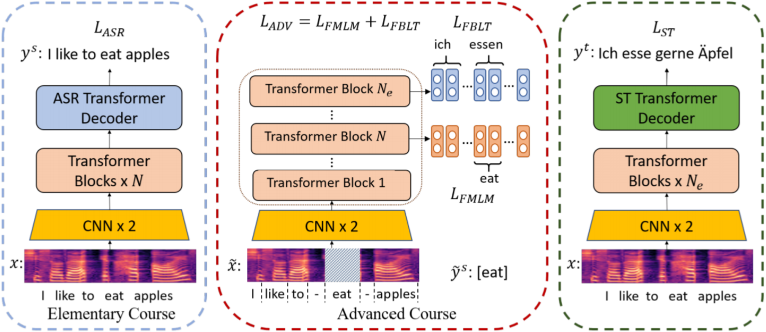
Require: $\mathscr{D}_{src}, \mathscr{D}_{tgt}$
Require: α, β: step size hyperparameters

Randomly initialize θ
while not done **do**
    Sample batch data $\mathcal{D}^{tra} \sim (\mathscr{D}_{src}, \mathscr{D}_{tgt})$, $\mathcal{D}^{val} \sim \mathscr{D}_{tgt}$
    for all $\mathcal{D}^{tra}_{\mathcal{T}_i} \in \mathcal{D}^{tra}$ **do**
        Evaluate $\nabla_\theta \mathcal{L}_{\mathcal{D}^{tra}_{\mathcal{T}_i}}(f_\theta)$ using $\mathcal{D}^{tra}_{\mathcal{T}_i}$
        Compute adapted parameters with gradient descent: $\theta'_{\mathcal{T}_i} = \theta - \alpha \nabla_\theta \mathcal{L}_{\mathcal{D}^{tra}_{\mathcal{T}_i}}(f_\theta)$
    end for
    $\theta \leftarrow \theta - \beta \sum_i \nabla_\theta \mathcal{L}_{\mathcal{D}^{val}}\left(f_{\theta'_{\mathcal{T}_i}}\right)$
end while

[1] From [14]

Curriculum Pre-training for End-to-end speech translation

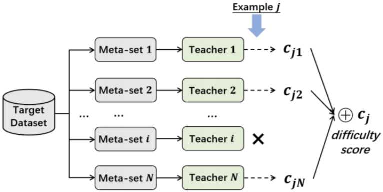
- Start using ASR
- Transition to predicting segments of audio based on layer.
- Add a decoder in final stage.



[1] [15]

Curriculum Learning for Natural Language Understanding

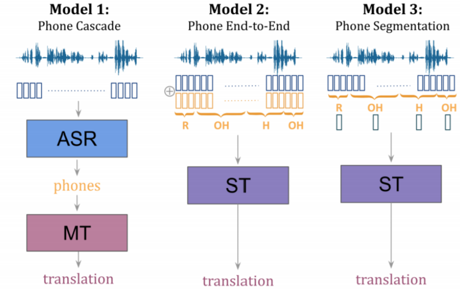
- Split up examples in training set into N meta-sets.
- Train a teacher model based on each of these sets.
- Score each training item (via Teachers)
- Sort training items into buckets
- Train by sampling from buckets, moving to harder buckets as training continues.



[1] [16]

Phone Features Improve Speech Translation

- Use seq2seq model to generate per-frame phone features (e.g. /R/)
- concat with audio and feed to ST model
- Results show 10 point BLEU increase with High resource setting (160hr) and 22 point increase with low setting (20hr)



[1] [17]

References I

[1] 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.

[2] 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.

[3] 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.

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.

References III

[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.

[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.

References IV

[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] N. Kitaev, Ł. Kaiser, and A. Levskaya, "Reformer: The efficient transformer," *arXiv preprint arXiv:2001.04451*, 2020.

[10] K. Irie, A. Zeyer, R. Schlüter, and H. Ney, "Language modeling with deep transformers," *arXiv preprint arXiv:1905.04226*, 2019.

References V

[11] 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.

[12] A. Bie, B. Venkitesh, J. Monteiro, M. Haidar, M. Rezagholizadeh *et al.*, "Fully quantizing a simplified transformer for end-to-end speech recognition," *arXiv preprint arXiv:1911.03604*, 2019.

References VI

[13] M. A. Di Gangi, M. Negri, and M. Turchi, "Adapting transformer to end-to-end spoken language translation," in *INTERSPEECH 2019*. International Speech Communication Association (ISCA), 2019, pp. 1133–1137.

[14] G. I. Winata, S. Cahyawijaya, Z. Lin, Z. Liu, P. Xu, and P. Fung, "Meta-transfer learning for code-switched speech recognition," *arXiv preprint arXiv:2004.14228*, 2020.

[15] C. Wang, Y. Wu, S. Liu, M. Zhou, and Z. Yang, "Curriculum pre-training for end-to-end speech translation," *arXiv preprint arXiv:2004.10093*, 2020.

References VII

[16] B. Xu, L. Zhang, Z. Mao, Q. Wang, H. Xie, and Y. Zhang, "Curriculum learning for natural language understanding," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 2020, pp. 6095–6104.

[17] E. Salesky and A. W. Black, "Phone features improve speech translation," *arXiv preprint arXiv:2005.13681*, 2020.