

Last Week

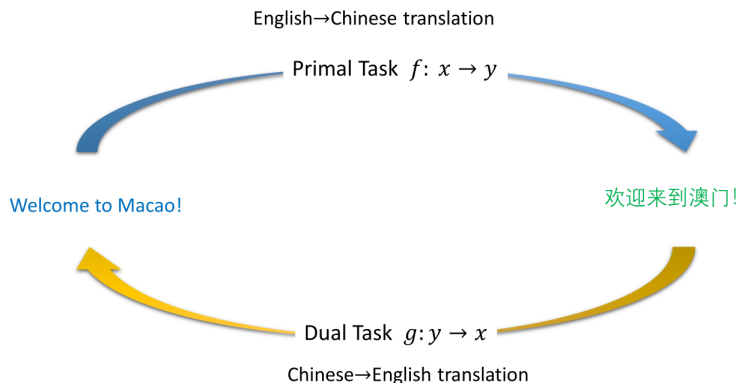


Figure: Dual Learning for Machine Translation. Taken from Xia et al. (2019).

Now

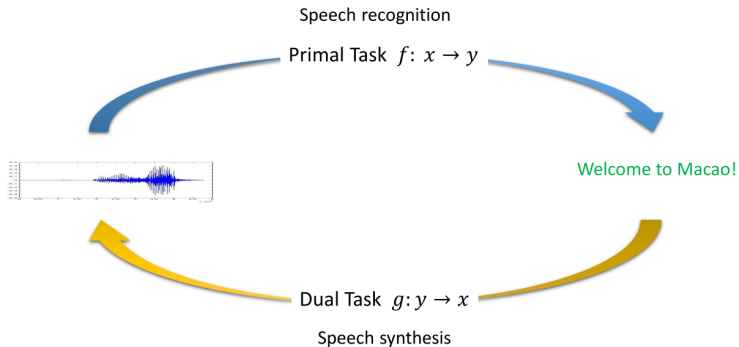


Figure: Dual Learning for Speech Processing. Taken from Xia et al. (2019).

Semi-Supervised Sequence-to-sequence ASR using Unpaired Speech and Text (Baskar et al. 2019)

Simon Will

Department of Computational Linguistics
Heidelberg University

Seminar: Recent Advances in Sequence-to-sequence Learning
Instructor: Tsz Kin Lam

January 23, 2020

Outline

Introduction

ASR and TTS Models

Cycle-Consistency Training

Speech Chain

Speech Chain with TTE

Baskar et al. 2019

Experiments of Baskar et al. (2019)

Conclusion

References

Questions

End-to-End Automatic Speech Recognition (E2E ASR)

In this paper: (Bi-)LSTM-based encoder-decoder setup.

Input Speech features $\vec{X} \in \mathbb{R}^{T \times d_x}$

Output Character sequence $\vec{Y} \in \mathbb{R}^C$

Encoder Output States $\vec{H} \in \mathbb{R}^{T \times d_h}$

Loss Cross-entropy loss

End-to-End Text-to-Speech (E2E TTS)

In this paper: Tacotron2 (Shen et al. 2018)

Input Character sequence $\vec{Y} \in \mathbb{R}^C$

Output Speech features $\vec{X} \in \mathbb{R}^{T \times d_x}$

Loss $L_{\text{TTS}} = L_{\text{MSE}} + L_{L_1} + L_{\text{BCE}}$

BCE Binary cross-entropy loss for EOS prediction

Tacotron2

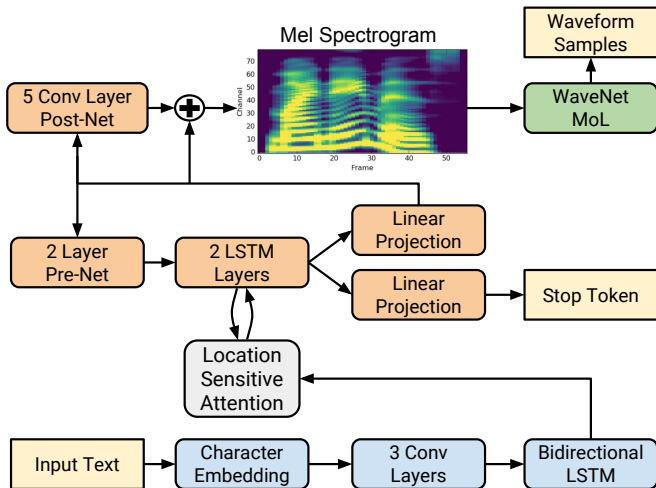


Figure: Architecture of Tacotron2. Diagram by Shen et al. (2018).

Speech Chain (Tjandra et al. 2017)

- ▶ Humans learn to speak and to understand at the same time
- Tjandra et al.: Let machines do the same
- ▶ Similar to Dual Learning for Machine Translation (He et al. 2016)

Speech Chain Algorithm (Tjandra et al. 2017)

- ▶ Use paired speech-text data to calculate losses L_{TTS}^P and L_{ASR}^P
- ▶ Use unpaired text and TTS to generate speech for ASR input, calculate loss L_{ASR}^U
- ▶ Use unpaired speech and ASR to generate text for TTS input, calculate loss L_{TTS}^U
- ▶ Final loss $L = \alpha(L_{\text{ASR}}^P + L_{\text{TTS}}^P) + \beta(L_{\text{ASR}}^U + L_{\text{TTS}}^U)$
- ▶ Calculate gradients for ASR and TTS separately

Speech Chain with Straight Through Estimator (Tjandra et al. 2019)

- ▶ Sampling from ASR's character distribution not differentiable
- ▶ Tjandra et al. (2017): Don't backpropagate further than nearest module
- ▶ Tjandra et al. (2019): Backpropagate via Gumbel softmax trick

Excursion: Gumbel Softmax Trick I

- ▶ Simultaneously introduced by Jang et al. (2017) and Maddison et al. (2017) building on Gumbel (1954)
- ▶ Sample $x \sim \mathcal{C}at(\pi)$ not differentiable
- ▶ Reparameterize:

$$\epsilon_i \sim \mathcal{G}umbel(0, 1) = -\ln(-\ln u_i) \quad u_i \sim \mathcal{U}niform(0, 1)$$
$$x = \text{one hot}(\arg \max_i (\epsilon_i + \ln \pi_i))$$

- ▶ Approximate $\arg \max$ with softmax with temperature T :

$$\tilde{x}_i = \frac{\exp(\frac{\epsilon_i + \log \pi_i}{T})}{\sum_j \exp(\frac{\epsilon_j + \log \pi_j}{T})}$$

- ▶ Use x or \tilde{x} for forward pass, but $\nabla \tilde{x}$ for backpropagation

Excursion: Gumbel Softmax Trick II

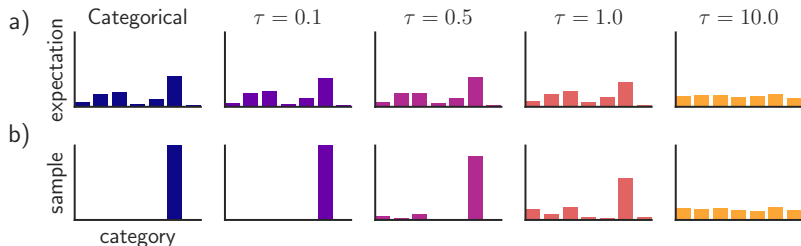


Figure: Categorical distribution and Gumbel softmax at different temperatures. Taken from Jang et al. (2017).

s/TTS/TTE/ (Hori et al. 2019)

- ▶ Problem with ASR output: No speaker characteristics and prosody information
- TTS *cannot* faithfully reconstruct original speech
 $\vec{X} \in \mathbb{R}^{T \times d_x}$
- ▶ Hori et al. (2019): Train Tacotron2 to reconstruct ASR encoder states $\vec{H} \in \mathbb{R}^{T \times d_h}$
- ▶ Disadvantage: No TTS model is trained
- Only usable for ASR training
- ▶ Use policy gradient to overcome backpropagation problem

Cycle-Consistency with TTE

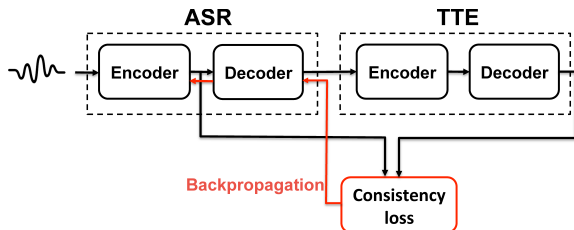


Figure: Cycle-consistency with TTE. Taken from Hori et al. (2019).

Explicitly Modeling Speaker Information

- ▶ Problems with TTE-CC:
 - ▶ Training useless TTE model
 - ▶ Encoded speech $\vec{H} \in \mathbb{R}^{T \times d_h}$ may already eliminate many speech characteristics
- Use TTS again, but incorporate speech characteristics via x-vectors (Snyder, Garcia-Romero, Sell, et al. 2018)
- ▶ With x-vector $f(\vec{X})$, Tacotron2 output probability changes from

$$p_{\text{TTS}}(\vec{X}^* | \vec{Y})$$

to

$$p_{\text{TTS}}(\vec{X}^* | \vec{Y}, f(\vec{X}))$$

Excursion: Speaker Verification and Recognition

“[E]ach speaker has his or her unique way of speaking, accent, pronunciation, pitch, rhythm, emotional state, etc. and there are differences even in the physical characteristics like vocal tract shapes or other sound production organs.” (Verma and Das 2015, p. 529)

- ▶ Can be used for “verifying” a speaker’s claimed identity
- ▶ Or for finding out (“recognizing”) a speaker’s identity
- ▶ Traditional approach: i-vectors (for “identity”) based on Gaussian-Mixture-Model-based Universal Background Model and Joint Factor Analysis

Excursion: x-Vectors as speaker embeddings

- ▶ Snyder, Garcia-Romero, Sell, et al. (2018) and Snyder, Garcia-Romero, Povey, et al. (2017)
- ▶ Train DNN to predict speaker from speech signal
- ▶ Use layer near end as embedding

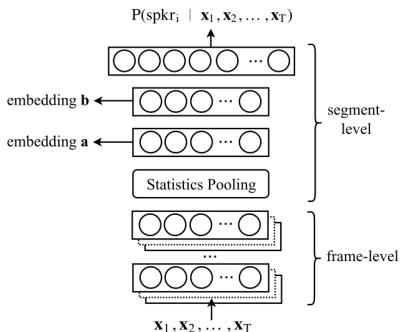


Figure: x-Vector DNN. Taken from Snyder, Garcia-Romero, Povey, et al. (2017).

CC Training with ASR and TTS

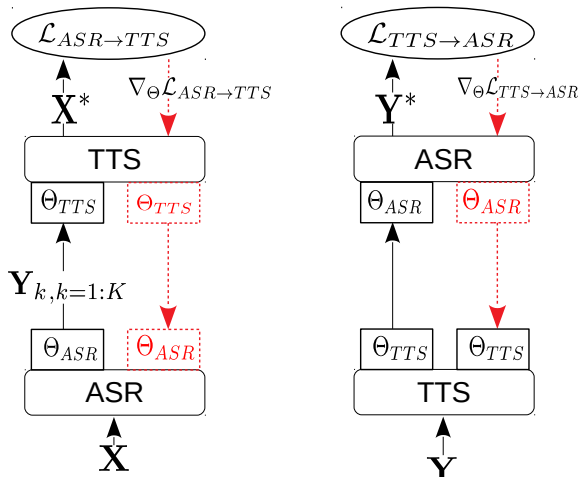
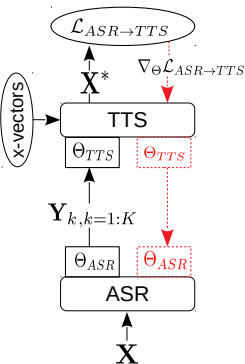


Figure: TTS-ASR-CC. Taken from Baskar et al. (2019)

ASR \rightarrow TTS



Use policy gradient (with bias $B(\vec{X})$) for backpropagation into ASR model

$$L_{ASR \rightarrow TTS} = \mathbb{E}_{p_{ASR}(\vec{Y}|\vec{X})} L_{TTS}$$

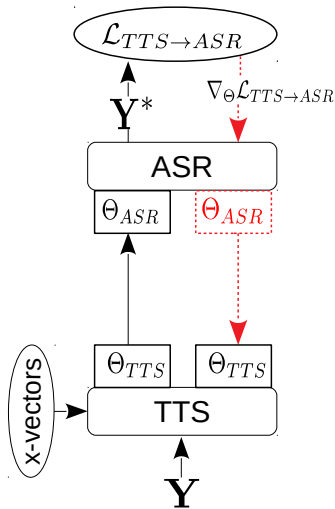
$$\nabla L_{ASR \rightarrow TTS} = \sum_{\vec{Y}^n \sim p_{ASR}} \frac{1}{N} R(\vec{Y}^n, \vec{X}) \nabla \log p_{ASR}(\vec{Y}^n | \vec{X})$$

$$R(\vec{Y}^n, \vec{X}) = L_{TTS} - B(\vec{X})$$

TTS \rightarrow ASR

- ▶ Goal only training ASR
- No need to backpropagate into TTS model

$$L_{TTS \rightarrow ASR} = -\log p_{ASR}(\vec{Y}^* | \vec{X})$$



Loss and Kind of Data

- ▶ With unpaired speech and text:

$$L_{\text{both}} = \alpha L_{\text{ASR} \rightarrow \text{TTS}} + (1 - \alpha) L_{\text{TTS} \rightarrow \text{ASR}}$$

Experimental Setup

- ▶ Librispeech and WSJ for training
- ▶ WSJ eval92 as test set
- ▶ 83-dimensional log-Mel filterbank as speech features
- ▶ Five samples from p_{ASR} (i.e., $N = 5$)
- ▶ Paired and unpaired data used for training

Paired Data Baseline

Name	Hours	% CER	% WER
WSJ	2	27.7	68.2
WSJ	5	13.2	41.5
WSJ	10	10.8	33.7
WSJ	14	10.2	31.5
Librispeech	100	8.9	21.0

Table: Results on WSJ eval-92 and Librispeech test-clean for different amounts of paired data.

Varying Amount of Paired Data

#hrs	Unpaired data Type	Paired data (#hrs)			
		2	5	10	14
14	Speech	49.8	39.9	29.8	-
14	Text	63.0	43.6	34.6	-
14	Both	43.7	35.5	28.3	-
67	Speech	51.9	38.8	28.4	28.0
67	Text	39.6	36.8	29.6	27.1
67	Both	41.4	34.2	27.7	26.2

Table: % WER on eval-92 for varying amounts of paired data.

Comparison with Other Systems

WSJ-SI84 (parallel) + WSJ-SI284 (unpaired)				
Model	Type	RNNLM	%CER	%WER
Speech chain [13]	Both	-	9.9	-
Adversarial [14]	Both	yes	-	24.9
this work	Both	-	9.1	26.1
this work	Both	yes	7.8	20.3
oracle	-	-	5.5	16.4
oracle [29]	-	yes	2.0	4.8
Librispeech 100 h (parallel) + 360 h (unpaired)				
Backtranslation-TTE [10]	Text	-	10.0	22.0
this work	Text	-	8.0	17.9
Criticizing-LM [12]	Text	yes	9.1	17.3
this work	Text	yes	8.0	17.0
Cycle-TTE [9]	Speech	yes	9.9	19.5
this work	Speech	yes	7.8	16.8
Adversarial-AE [15]	Both	yes	8.4	18.0
this work	Both	-	7.6	17.5
this work	Both	yes	7.6	16.6
oracle [9]	-	-	4.6	11.8

Table: Results of different systems in the literature. Only unpaired data used (except oracles).

Critique

- ▶ Good overview of existing CC methods
- ▶ Overall plausible results and improvement over previous models
- ▶ Lack of plausible theories for how the results come about
- ▶ What about loss of variance of speech inputs?

References I



Baskar, Murali Karthick et al. (2019). “Semi-supervised Sequence-to-sequence ASR using Unpaired Speech and Text”. In: *Interspeech 2019* (Interspeech), pp. 3790–3794. URL: <http://dx.doi.org/10.21437/Interspeech.2019-3167>.



Dehak, Najim et al. (2011). “Front-End Factor Analysis for Speaker Verification”. In: *IEEE Transactions on Audio, Speech, and Language Processing* 19.4, pp. 788–798.



Gumbel, Emil Julius (1954). *Statistical Theory of Extreme Values and Some Practical Applications*. A Series of Lectures. Washington: U.S. Government Printing Office.







He, Di et al. (2016). “Dual Learning for Machine Translation”. In: *Advances in Neural Information Processing Systems 29* (NIPS), pp. 820–828. URL: <https://papers.nips.cc/paper/6469-dual-learning-for-machine-translation>.







Hori, Takaaki et al. (2019). “Cycle-consistency Training for End-to-end Speech Recognition”. In: *2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 6271–6275.

References II

-  Jang, Eric, Shixiang Gu, and Ben Poole (2017). “Categorical Reparameterization with Gumbel-Softmax”. In: *Proceedings of the 5th International Conference on Learning Representations (ICLR)*.
-  Maddison, Chris J., Andriy Mnih, and Yee Whye Teh (2017). “The Concrete Distribution: A Continuous Relaxation of Discrete Random Variables”. In: *Proceedings of the 5th International Conference on Learning Representations (ICLR)*.
-  Shen, Jonathan et al. (2018). “Natural TTS Synthesis by Conditioning WaveNet on Mel Spectrogram Predictions”. In: *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 4779–4783.
-  Snyder, David, Daniel Garcia-Romero, Daniel Povey, et al. (2017). “Deep Neural Network Embeddings for Text-Independent Speaker Verification”. In: *Interspeech 2017 (Interspeech)*, pp. 999–1003.

References III

-  Snyder, David, Daniel Garcia-Romero, Gregory Sell, et al. (2018). “X-Vectors: Robust DNN Embeddings for Speaker Recognition”. In: *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 5329–5333.
-  Tjandra, Andros, Sakriani Sakti, and Nakamura Satoshi (2017). “Listening while speaking: Speech chain by deep learning”. In: *2017 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pp. 301–308.
-  – (2019). “End-to-end Feedback Loss in Speech Chain Framework via Straight-through Estimator”. In: *2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 6281–6285.
-  Verma, Pulkit and Pradip K. Das (2015). “i-Vectors in speech processing applications: a survey”. In: *International Journal of Speech Technology* 18, pp. 529–546.
-  Xia, Yingce et al. (2019). *Dual Learning*. URL: <https://www.microsoft.com/en-us/research/project/dual-learning/>.

Questions I

Question

How is the loss $L_{\text{ASR} \rightarrow \text{TTE}}$ end-to-end differentiable? This question is effectively aimed at the straight-through estimation defined in reference 8 (Tjandra et al. 2019). Why is this approximation valid?

Answer

Baskar et al. (2019) use policy gradient, Tjandra et al. (2019) use straight-through-Gumbel-softmax approximation.

Questions II

Question

“A central problem to the ASR- \rightarrow TTS pipeline is the fact that the text bottleneck eliminates a lot of information from speech e.g. speaker identity.” Why is preserving “information from speech” desirable?

Answer

If it is lost, the TTS cannot reasonably be expected to accurately reconstruct the original speech because information is missing.

Questions III

Question

I found that there was quite a lack of qualitative results. Since the code is openly available, do they somewhere provide qualitative results/examples?

Answer

Didn't find the code.

Questions IV

Question

What is WER and CER? Maybe it is something obvious I am not grasping, but I did not find an explanation in the paper.

Answer

Character Error Rate, Word Error Rate.

Questions V

Question

What is the Tacotron2 architecture and why was it chosen?

Answer

Shown above. Seems to be popular. Also, it provides mel-spectrograms as opposed to directly outputting a waveform

Questions VI

Question

What is a x-vector network?

Answer

Explained above.

Questions VII

Question

Section 4.2: It is stated that the performance on WSJ-S1284 is inferior to the previously reported baseline due to architecture changes to fit the model into the GPU. Question if you have taken a look into their github codebase (or this is written somewhere in the paper and I missed it): Any idea what these changes were and why they reduced the performance to this extent (4.8 to 20.3 seems to be a quite large gap)?

Answer

???

Questions VIII

Question

The authors state that “the text bottleneck eliminates a lot of information from speech e.g. speaker identity”. Not having a lot of experience in speech processing: What exactly does speaker identity mean - voice, intonation? How can this be modeled through vectors? Also: “e.g. speaker identity”, what else could be an issue? (Section 2.2)

Answer

Shown above.

Questions IX

Question

I am confused about the TTE approach: So you compare encodings H to \hat{H} , right? Thus, I understand that you start with a text and only model the encodings H , but not the final speech sequence - Then where does the ASR system come from which apparently transforms from speech back to encodings \hat{H} for comparison? Do you still generate the speech, but just don't compare at speech level? If so, would the encodings \hat{H} not still carry speaker characteristics "on the way back"?

Answer

TTE-CC can only be used for starting with the speech signal, not for starting with some text.

Questions X

Question

what is the concrete trick to propagate the gradient from TTS to ASR? Do they have the same target?

Answer

Policy gradient for $L_{ASR \rightarrow TTS}$, normal backpropagation with cross-entropy for $L_{TTS \rightarrow ASR}$

Questions XI

Question

After Eq. (9) the authors state: “Note that x-vectors are designed to retain speaker characteristics but not general structure of the speech signal. In that sense, the model can not learn to copy directly X from input to output.” What does this mean? Is $f(X)$ used as kind of regularization?

Answer

Not a regularization. It's modeling speech characteristics that are present in the ASR input, but not its output.

Questions XII

Question

What color does a Smurf turn if you choke it?

Answer

I assume, it depends on their latent ethnicity. E.g. white for caucasian smurfs, red for native American smurfs, etc.

Questions XIII

Question

What can we understand under a pitch feature?

Answer

Different values for deep and high voice?

Questions XIV

Question

What are some speaker characteristics and how are they represented?

Answer

Shown above.

Questions XV

Question

Section 4.1: Why does performance drop with high amounts of unpaired speech, but not with high amounts of unpaired text?

Answer

???