IFT 4030/7030, Machine Learning for Signal Processing

Week12: Speech/Audio

Cem Subakan

Admin

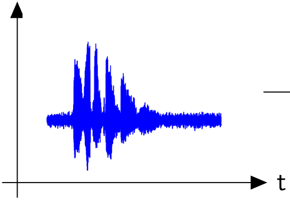
- Les labos sont finis!
  - We are finally done with the labs!
- Je vais publier le troisième devoir cette semaine. C'est optionnel.
  - I'll release the third homework this week. It will be optional.
- N'oubliez pas d'inscrire votre projet sur le fichier excel:
  - Don't forget to sign-up for a project presentation slot! `https://docs.google.com/spreadsheets/d/1uZYn_RLkZ_CpQxXTRgoRwZ6b8PdrUtWsZTI1aD-L7Hs/edit?usp=sharing`
- Le formulaire pour les évaluations du cours est publié. C'est envoyé à votre courriel.
  - The class evaluation form is out. It's sent to your email.
- Cette semaine, on a le dernier cours, et c'est sur speech et audio!
  - This week is the last class, and it's on speech and audio!

Today

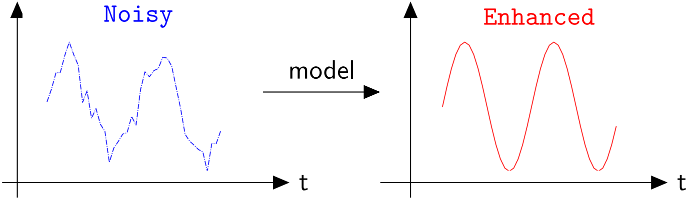
- ASR, TTS, Speech Separation / Enhancement, Text-Audio Representations, Interpretability
- The first part will be more like our usual classes. / La premiere partie va etre comme business as usual.
- It will get more like a research talk afterwards. / Je vais parler plus comme si c'était une presentation de recherche dans la deuxième partie.

Typical Applications in Speech and Audio Modeling

- Speech Recognition

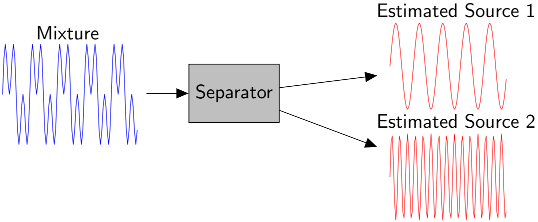


- Speech Enhancement

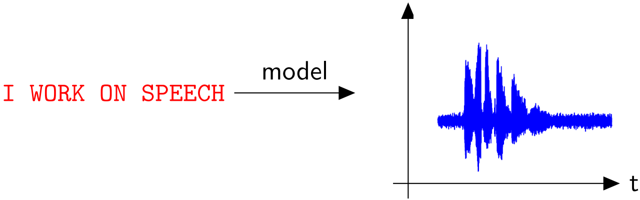


Speech and Audio Modeling

- Speech Separation

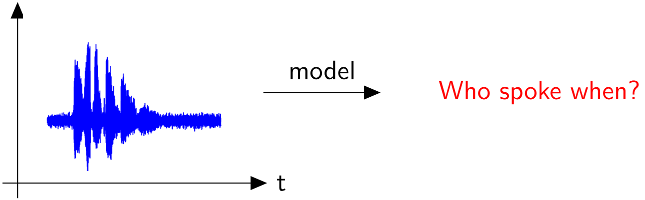


- Text-to-Speech

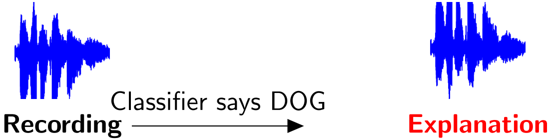


Speech and Audio Modeling

- Speaker Diarization



- Neural Network Explanation



- Other problems: Generating Deep fakes, Detecting deep fakes, Music Source Separation, Music Transcription, Sound Event Detection/Classification...

Table of Contents

Automatic Speech Recognition

- In Automatic Speech Recognition (ASR) the goal is to convert a an audio signal into text.
  - Dans la tache reconnaissance vocale le but est de conventir un signal audio au texte.
- We already implemented a very simple ASR system in this class by the way!
  - On a déjà fait un exercise dans le lab1 pour un système d'ASR simple.

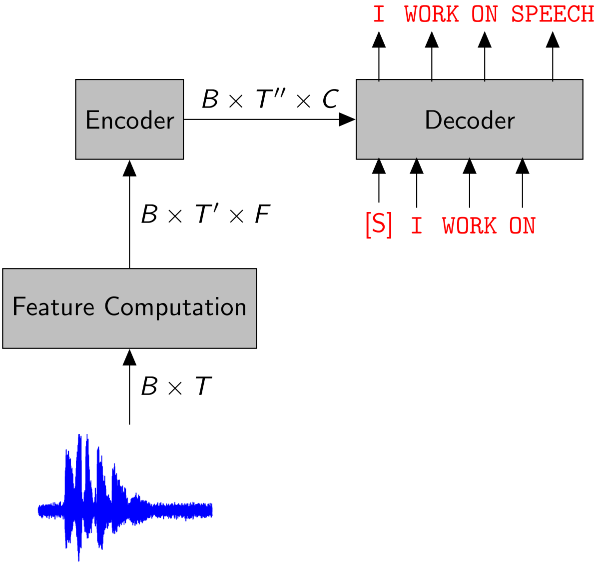
Automatic Speech Recognition

- In Automatic Speech Recognition (ASR) the goal is to convert a an audio signal into text.
  - Dans la tache reconnaissance vocale le but est de conventir un signal audio au texte.
- We already implemented a very simple ASR system in this class by the way!
  - On a déjà fait un exercise dans le lab1 pour un système d'ASR simple.
- We also talked about HMMs which used to be the SOTA for speech recognition in the 90s - 2000s.
  - On a aussi parlé des HMMs qui était l'approche SOTA pour la reconnaissance vocale dans les 90s - 2000s.

Automatic Speech Recognition

- In Automatic Speech Recognition (ASR) the goal is to convert a an audio signal into text.
  - Dans la tache reconnaissance vocale le but est de conventir un signal audio au texte.
- We already implemented a very simple ASR system in this class by the way!
  - On a déjà fait un exercise dans le lab1 pour un système d'ASR simple.
- We also talked about HMMs which used to be the SOTA for speech recognition in the 90s - 2000s.
  - On a aussi parlé des HMMs qui était l'approche SOTA pour la reconnaissance vocale dans les 90s - 2000s.
- Today we will talk about more modern approaches.
  - On va parler des approches modernes.

Encoder-Decoder Sequence-to-Sequence Learning



Feature Computation Block

- We turn an input waveform $x \in \mathbb{R}^T$, into a time-frequency representation $x' \in \mathbb{R}^{T' \times F}$.
  - On transforme un waveform x à une representation x' temps-fréquence.

Feature Computation Block

- We turn an input waveform $x \in \mathbb{R}^T$, into a time-frequency representation $x' \in \mathbb{R}^{T' \times F}$.
  - On transforme un waveform x à une representation x' temps-fréquence.
- ASR systems typically use mel-transformed spectra
  - Les systèmes d'ASR typiquement utilisent des spectrogrames en échelle-mel

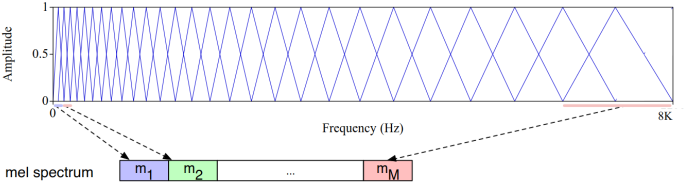


Image taken from Speech and Language Processing, Jurafsky, Martin

Table of Contents

The Encoder Block

- What we typically do in the encoder block is to reduce the sampling rate, while applying convolutions / RNNs.
  - On typiquement reduit le taux d'echantillon avec des convolutions / RNNs.

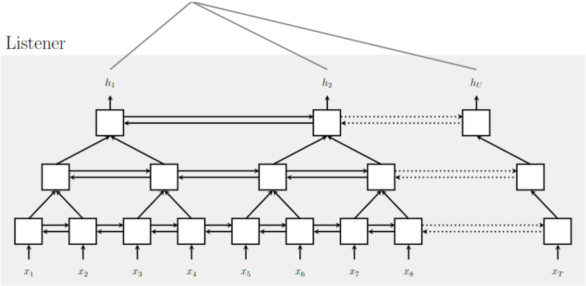


Image taken from the paper, Listen, Attend, Spell https://arxiv.org/pdf/1508.01211.pdf

- We can denote $h =: \text{Encoder}(x)$.

The Decoder Block

- The goal is to maximize the following probability distribution / Le but est de maximiser la distribution suivante,

$$\max \sum_{t=1}^{T} \log p(y_t | y_{t-1}, \ldots, y_1, x)$$

The Decoder Block

- The goal is to maximize the following probability distribution / Le but est de maximiser la distribution suivante,

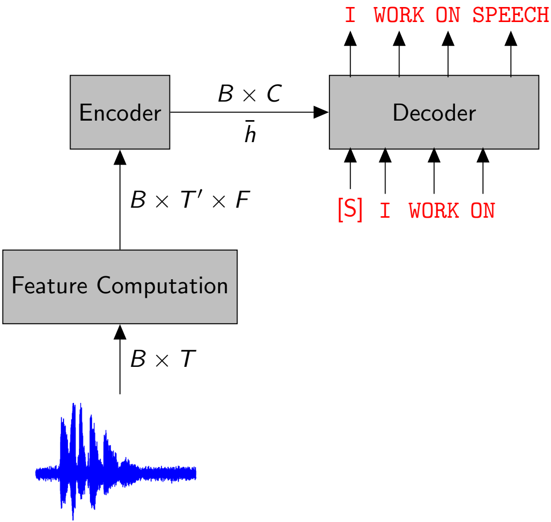
$$\max \sum_{t=1}^{T} \log p(y_t | y_{t-1}, \ldots, y_1, x)$$

- Here's a naive way of doing it / Une facon naive pour le faire:

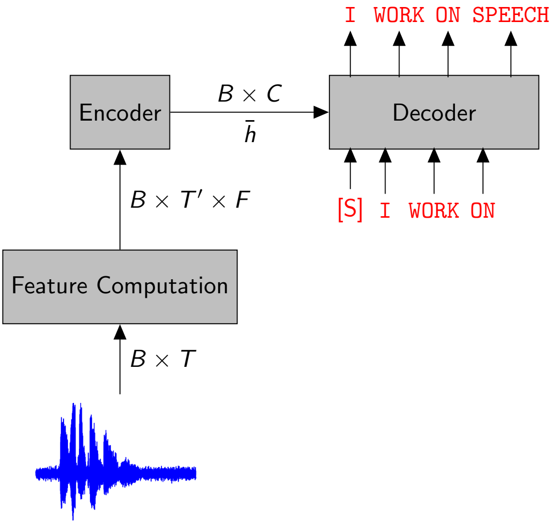
$$\begin{aligned}\bar{h} =& \frac{1}{T} \sum_t h_t \\ \widehat{y}_t =& \text{RNN}(y_{1:t-1}, \bar{h}) = \text{RNN}(y_{t-1}, s_{t-1}, \bar{h}\end{aligned}$$

We inject the average representation h to the autoregressive RNN model. / On injecte une representation average au modele autoregressive.

Naive encoder-decoder



Naive encoder-decoder



This system however shrinks the input signal resolution too much. / On perds trop de resolution temporelle avec ce système.

Encoder-Decoder with Attention in between

- The additional attention block / le bloque d'attention:

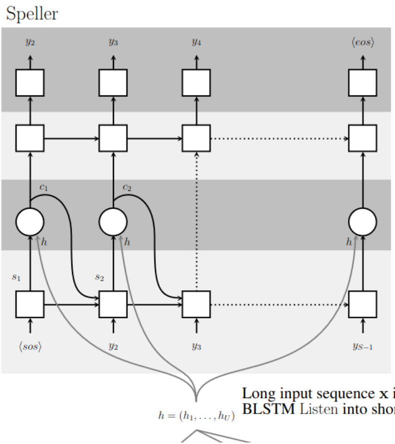
$$\begin{aligned} c_i =& \text{AttentionContext}(s_i, h_{1:T''}) \\ s_i =& \text{RNN}(s_{i-1}, y_{i-1}, c_{i-1}) \\ p(y_i|x, y_{1:i-1}) =& \text{OutputDistribution}(s_i, c_i) \end{aligned}$$

- Here's how it works / Voici comment ça fonctionne:

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

- $\phi(.)$, $\psi(.)$ are MLPs.

Listen, Attend, Spell Decoder



Grapheme characters y_i are modelled by the CharacterDistribution

AttentionContext creates context vector c_i from **h** and s_i

Long input sequence **x** is encoded with the pyramidal BLSTM Listen into shorter sequence **h**

Listen, Attend, Spell All picture

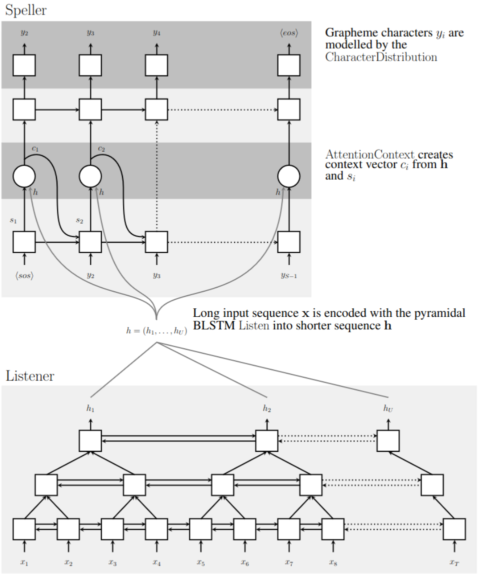


Table of Contents

Query-Key-Value Attention

- Attention is the basis of the transformer architecture which obtains state-of-the art results in several domains such as NLP, computer vision, speech recognition. / Attention est base de l'architecture transformer qui obtient SOTA dans plusieurs domains.

$$\text{Attention}(X_1, X_2, X_3) = \text{softmax}\left(\frac{QK^\top}{\sqrt{d_k}}\right) V$$

$$Q = X_1 W^Q,\ K = X_2 W^K,\ V = X_3 W^V$$

Query-Key-Value Attention

- Attention is the basis of the transformer architecture which obtains state-of-the art results in several domains such as NLP, computer vision, speech recognition. / Attention est base de l'architecture transformer qui obtient SOTA dans plusieurs domains.

$$\text{Attention}(X_1, X_2, X_3) = \text{softmax}\left(\frac{QK^\top}{\sqrt{d_k}}\right) V$$

$$Q = X_1 W^Q,\ K = X_2 W^K,\ V = X_3 W^V$$

$$\begin{bmatrix} \sum_j a_{0j} v_j \\ \sum_j a_{1j} v_j \\ \sum_j a_{2j} v_j \\ \sum_j a_{3j} v_j \end{bmatrix} = \begin{bmatrix} a_{0,0} & a_{0,1} & a_{0,2} & a_{0,3} \\ a_{1,0} & a_{1,1} & a_{1,2} & a_{1,3} \\ a_{2,0} & a_{2,1} & a_{2,2} & a_{2,3} \\ a_{3,0} & a_{3,1} & a_{3,2} & a_{3,3} \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \\ v_3 \\ v_4 \end{bmatrix}$$

Query-Key-Value Attention

- Attention is the basis of the transformer architecture which obtains state-of-the art results in several domains such as NLP, computer vision, speech recognition. / Attention est base de l'architecture transformer qui obtient SOTA dans plusieurs domains.

$$\text{Attention}(X_1, X_2, X_3) = \text{softmax}\left(\frac{QK^\top}{\sqrt{d_k}}\right) V$$

$$Q = X_1 W^Q,\ K = X_2 W^K,\ V = X_3 W^V$$

$$\begin{bmatrix} v_3 \\ \frac{v_3}{2} + \frac{v_4}{2} \\ v_2 \\ v_1 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0.5 & 0.5 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \\ v_3 \\ v_4 \end{bmatrix}$$

Multi-head Attention

- We calculate parallel attentions, and then combine them. / On calcule des attentions parallèles, et les combine.

$$\mathrm{MHA}(X_1, X_2, X_3) = \mathrm{Concatenate}(\mathrm{Attention}_1, \ldots, \mathrm{Attention}_h) W^O$$

$$\text{where } \mathrm{Attention}_i = \mathrm{softmax}\left(\frac{X_1 W_i^Q (X_2 W_i^K)^\top}{\sqrt{d_k}}\right) X_3 W_i^V$$

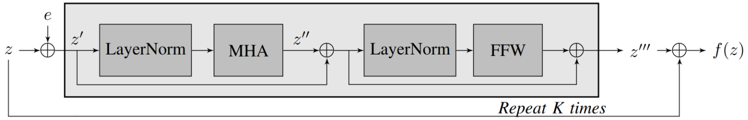
$$X_1, X_2, X_3 \in \mathbb{R}^{T \times L},\ W^Q, W^K, W^V \in \mathbb{R}^{L \times L}$$

Self-Attention and the Transformer Encoder

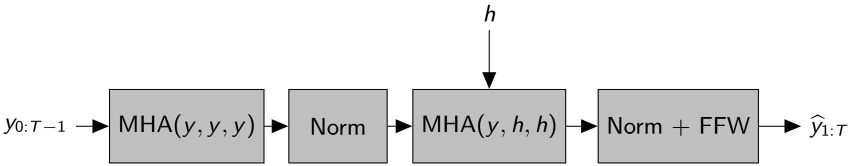
- If the inputs are the same sequence, this is called self-attention / Si les séquences d'entrée sont les memes, on appelle ça self-attention

$$\text{Self-Attention}(X) = \text{MHA}(X, X, X)$$

- In addition to Multihead attention, we also have, normalizations, a feed-forward layer, positional embeddings, and skip connections. / On a aussi des normalisations, des connections qui sautent, et une couche feed-forward, et les embeddings positionnel.



The Transformer Decoder



- I am ignoring the skip connections for simplicity. / Je n'iclus pas les connections qui sautent pour la simplicité dans la figure.
- The idea is very simple to RNN with attention, but we do it via the multi-head attention layers.

Replacing the Attention Layer with Multi-Head Attention

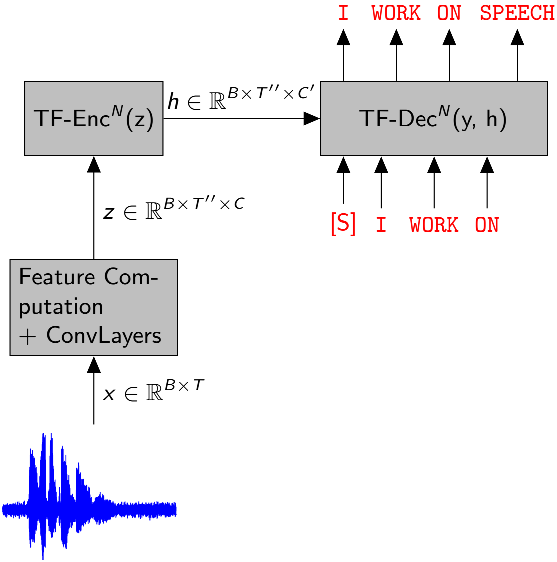
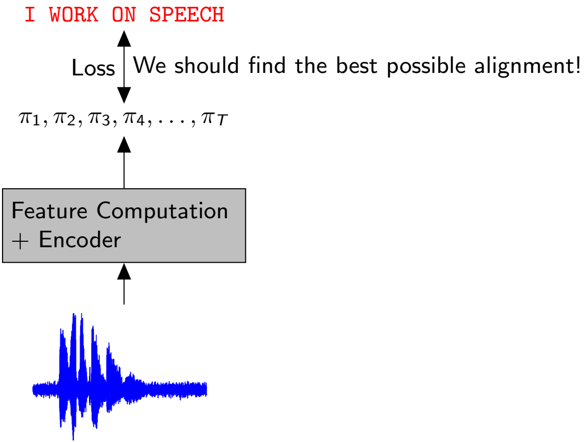


Table of Contents

Another Approach for Alignment

- ASR is a sequence-to-sequence problem. We need to resolve an alignment between the network output, and the target sequence. / ASR est un problème de sequence-à sequence. Il faut résoudre l'alignement entre la sortie du network, et la séquence des characteres à la sortie.
- We saw the encoder - decoder architecture. Decoder takes care of the alignment. / Le decoder trouve une resolution pour ce problème.



Naive alignment approach

- Maybe we can just merge the repeated characters? / Peut-etre on peut merger les characteres qui répetent?

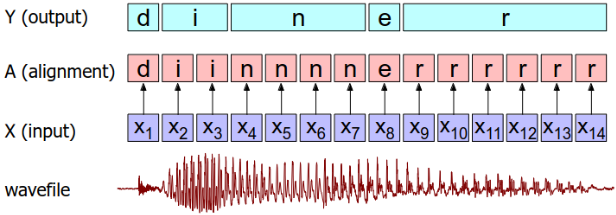


Image taken from Speech and Language Processing, Jurafsky, Martin

Naive alignment approach

- Maybe we can just merge the repeated characters? / Peut-etre on peut merger les characteres qui répetent?

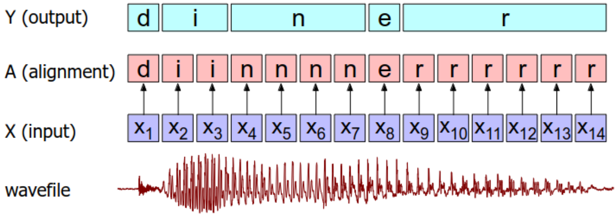


Image taken from Speech and Language Processing, Jurafsky, Martin

- What's wrong with this? / Qu'est-ce qui ne fonctionne pas içi?

Naive alignment approach

- Maybe we can just merge the repeated characters? / Peut-etre on peut merger les characteres qui répetent?

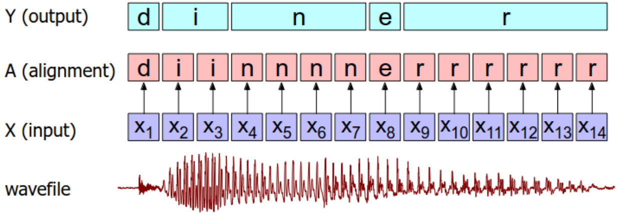


Image taken from Speech and Language Processing, Jurafsky, Martin

- What's wrong with this? / Qu'est-ce qui ne fonctionne pas içi?
- Dinner vs diner ??? , silences?

Connectionist Temporal Classification

- We can however find alignments by inserting a separator character. / On peut trouver un alignment en trouvant un meilleur alignment.

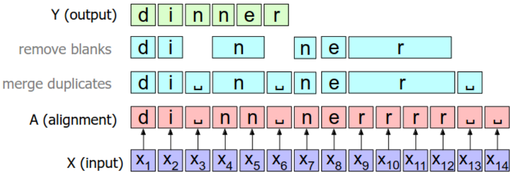


Image taken from Speech and Language Processing, Jurafsky, Martin

Connectionist Temporal Classification

- We can however find alignments by inserting a separator character. / On peut trouver un alignment en trouvant un meilleur alignment.

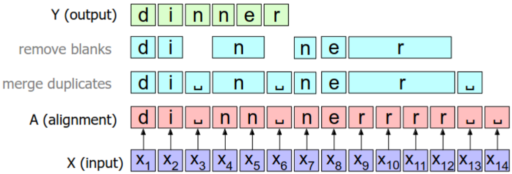


Image taken from Speech and Language Processing, Jurafsky, Martin

- How do we find this alignment though? Alignments which would work for the example above. / Comment trouve-t-on un alignment? Ces alignments suivants fonctionnerait par exemple pour l'exemple en haut:

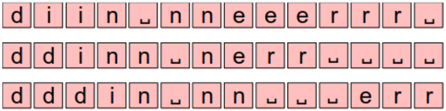


Image taken from Speech and Language Processing, Jurafsky, Martin

Finding the best possible alignment

- Training objective:

$$\max_{\theta} p(y_{1:T}|\pi_{1:T},\theta) = \max_{\theta} \sum_{A_{1:T}} p(y_{1:T}, A_{1:T}|\pi_{1:T},\theta)$$

- The CTC assumes a Markov chain on $A_{1:T}$ / CTC suppose un chaine Markov:

$$p(y_{1:T}, A_{1:T}) = \prod_t p(y_t|A_t)p(A_{t+1}|A_t) = \prod_t \pi_{A_t} p(A_{t+1}|A_t)$$

Finding the best possible alignment

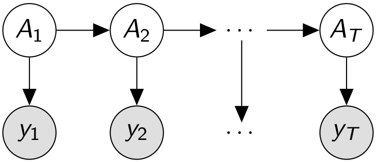
- Training objective:

$$\max_\theta p(y_{1:T}|\pi_{1:T},\theta) = \max_\theta \sum_{A_{1:T}} p(y_{1:T}, A_{1:T}|\pi_{1:T},\theta)$$

- The CTC assumes a Markov chain on $A_{1:T}$ / CTC suppose un chaine Markov:

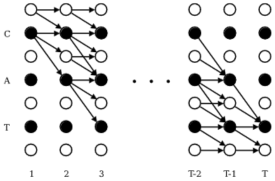
$$p(y_{1:T}, A_{1:T}) = \prod_t p(y_t|A_t)p(A_{t+1}|A_t) = \prod_t \pi_{A_t} p(A_{t+1}|A_t)$$

- This is an HMM! (with a specific transition structure) / C'est un HMM avec une structure de transition specifique.



Emission Model: $p(y_t|A_t) = \pi_{A_t}$

Transition Model:



How do we calculate CTC then?

- We need to calculate / On doit calculer:

$$p(y_{1:T}) = \sum_{A_{1:T}} p(y_{1:T}, A_{1:T})$$

$$\alpha(A_t) = p(y_t|A_t) \sum_{A_{t-1}} p(A_t|A_{t-1}) \underbrace{p(y_{t-1}|A_{t-1}) \dots \underbrace{p(y_2|A_2) \sum_{A_1} p(A_2|A_1) p(y_1|A_1) \underbrace{p(A_1)}_{\alpha(A_1)}}_{\alpha(A_2)}}_{\alpha(A_{t-1})}$$

$$p(y_{1:T}) = \sum_{A_T} \alpha(A_T)$$

How do we calculate CTC then?

- We need to calculate / On doit calculer:

$$p(y_{1:T}) = \sum_{A_{1:T}} p(y_{1:T}, A_{1:T})$$

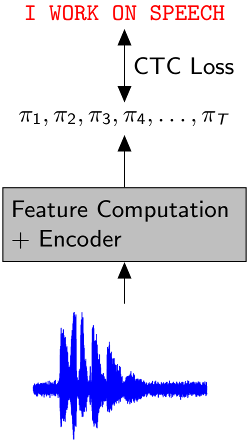
- Dynamic Programming to the rescue (the forward pass we talked about two weeks ago) / Programmation Dynamique va nous sauver. (la recurrence en avancant dont on a parlé il y a deux semaines)

$$\alpha(A_t) = p(y_t|A_t) \sum_{A_{t-1}} p(A_t|A_{t-1}) \underbrace{p(y_{t-1}|A_{t-1}) \ldots \underbrace{p(y_2|A_2) \sum_{A_1} p(A_2|A_1) p(y_1|A_1) \underbrace{p(A_1)}_{\alpha(A_1)}}_{\alpha(A_2)}}_{\alpha(A_{t-1})}$$

$$p(y_{1:T}) = \sum_{A_T} \alpha(A_T)$$

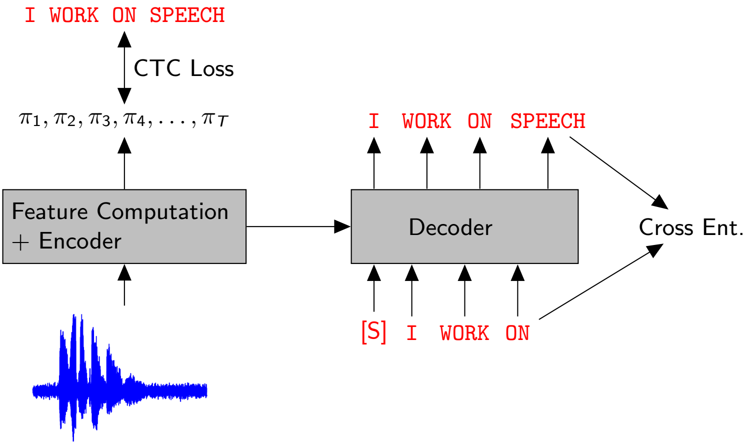
Of course you can also just call `torch.nn.CTCLoss`. :)

CTC Based ASR



CTC + Encoder-Decoder

- It's also possible to use both CTC and a decoder! / On peut aussi utiliser CTC et un decodeur.



Incorporating a Language Model

- In addition to the probability $p(y_{1:T}|X)$, we can also include a language model score $\mathcal{LM} = \prod_t p(y_t|y_{1:t-1})$. The final score function when decoding is then as follows:
  - On peut aussi incorporer un language model quand on fait le decodage. Dans ce-cas ci le score final est comme le suivant:

$$\widehat{Y} = \arg\max_{Y} [\lambda \log p_{encdec}(Y|X) + (1-\lambda) \log p_{CTC}(Y|X) + \gamma p_{LM}(Y)]$$

Performance Evaluation

- Word-Error Rate:

$$WER = 100 \times \frac{Insertions + Substituitions + Deletions}{Total\ N.Words}$$

REF:	i	***	**	UM	the	PHONE	IS	i	LEFT	THE	portable	****	PHONE	UPSTAIRS	last	night
HYP:	i	GOT	IT	TO	the	*****	FULLEST	i	LOVE	TO	portable	FORM	OF	STORES	last	night
Eval:		I	I	S		D	S		S	S		I	S	S		

This utterance has six substitutions, three insertions, and one deletion:

$$\text{Word Error Rate} = 100\frac{6+3+1}{13} = 76.9\%$$

Image taken from Speech and Language Processing, Jurafsky, Martin

Typical Performance under LibriSpeech

- LibriSpeech is a 16kHz speech dataset with over 1000 hours of audio books. Sentences are aligned at the sentence level. / LibriSpeech est un jeu de données large qui a au délà de 1000 heures. Les phrases sont alignés de niveau phrase.
    - The transformer model in SpeechBrain obtains 2 % WER on test-clean.
    - With a wav2vec pretrained encoder trained with CTC, we obtain 1.9 % WER.
    - An RNN based encoder-decoder model obtains 3 $ WER.
- If you want to check yourself / Si vous voulez voir vous meme `https://github.com/speechbrain/speechbrain/tree/develop/recipes/LibriSpeech`

Real-life Generalization

- Deep learning methods have been shown to work extremely well on synthetic benchmarks.
  - $\sim$ 2% Word-error rate on LibriSpeech test set.
  - $>$20dB SNR for speech separation on WSJ0-2Mix.
  - $>$3 PESQ for speech enhancement on Voicebank.
- It is not clear that these models generalize well on real-data.
  - **Recording 1:**

Real-life Generalization

- Deep learning methods have been shown to work extremely well on synthetic benchmarks.
  - $\sim$ 2% Word-error rate on LibriSpeech test set.
  - $>$20dB SNR for speech separation on WSJ0-2Mix.
  - $>$3 PESQ for speech enhancement on Voicebank.
- It is not clear that these models generalize well on real-data.
  - **Recording 1:**
  - `HE'LL BEEN RUTH MORAL` (WER 3.0 on LS)

Real-life Generalization

- Deep learning methods have been shown to work extremely well on synthetic benchmarks.
  - $\sim$ 2% Word-error rate on LibriSpeech test set.
  - $>$20dB SNR for speech separation on WSJ0-2Mix.
  - $>$3 PESQ for speech enhancement on Voicebank.
- It is not clear that these models generalize well on real-data.
  - **Recording 1:**
  - `HE'LL BEEN RUTH MORAL` (WER 3.0 on LS)
  - `PESTEIN THIS MORAL` (WER 2.2 on LS)

Real-life Generalization

- Deep learning methods have been shown to work extremely well on synthetic benchmarks.
  - $\sim$ 2% Word-error rate on LibriSpeech test set.
  - >20dB SNR for speech separation on WSJ0-2Mix.
  - >3 PESQ for speech enhancement on Voicebank.
- It is not clear that these models generalize well on real-data.
  - **Recording 1:**
  - `HE'LL BEEN RUTH MORAL` (WER 3.0 on LS)
  - `PESTEIN THIS MORAL` (WER 2.2 on LS)
  - `TESTING THIS MOLO` (pretr. wav2vec2 backbone, fine-tuned on CommonVoice)

Real-life Generalization

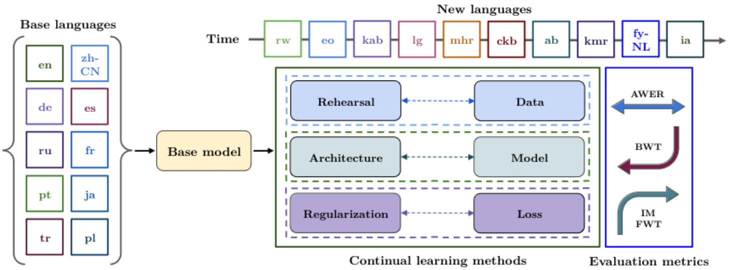
- Deep learning methods have been shown to work extremely well on synthetic benchmarks.
  - $\sim$ 2% Word-error rate on LibriSpeech test set.
  - >20dB SNR for speech separation on WSJ0-2Mix.
  - >3 PESQ for speech enhancement on Voicebank.
- It is not clear that these models generalize well on real-data.
  - **Recording 1:**
  - `HE'LL BEEN RUTH MORAL` (WER 3.0 on LS)
  - `PESTEIN THIS MORAL` (WER 2.2 on LS)
  - `TESTING THIS MOLO` (pretr. wav2vec2 backbone, fine-tuned on CommonVoice)

  - **Recording 2:**
  - `HE WAS MALLOW` (WER 3.0 on LS)
  - `PESSIM WAS MODEL` (WER 2.2 on LS)
  - `TESTING THIS MODEL` (pretr. wav2vec2 backbone, fine-tuned on CommonVoice)

Continual Learning of Multi-Lingual ASR Models

CL-MASR: A Continual Learning Benchmark for Multilingual ASR

Luca Della Libera*, Pooneh Mousavi*, Salah Zaiem, Cem Subakan, Mirco Ravanelli



The paper: https://arxiv.org/pdf/2310.16931.pdf

Table of Contents

Tacotron

- Again a sequence-to-sequence architecture / Encore une fois une architecture séquence-à-séquence

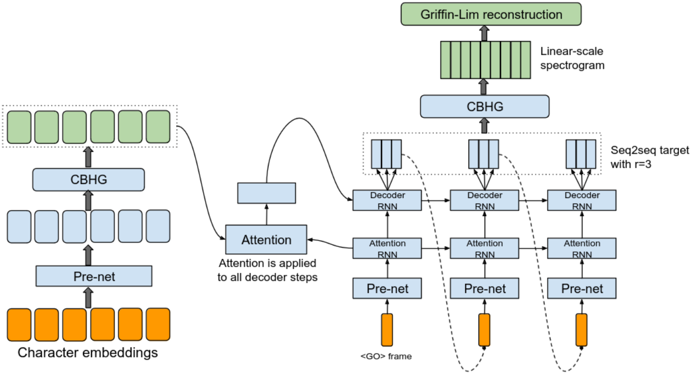


Image taken from the original paper `https://arxiv.org/pdf/1703.10135.pdf`

A typical modern TTS pipeline

- First sequence of characters is converted into a series of representations (encoder) / L'encodeur transforme la texte à une serie de vecteurs.
- Then, a decoder, with the help of an attention mechanism, predicts the next mel-spectrogram column. / Le decodeur avec l'aide d'une mechanisme d'attention prédit la colonne suivante d'un mel-spectrogramme.
- The mel-spectrogram is turned into audio with a vocoder. / Le mel-spectrogramme est transformé à l'audio en utilisant un vocoder.

Tacotron 2

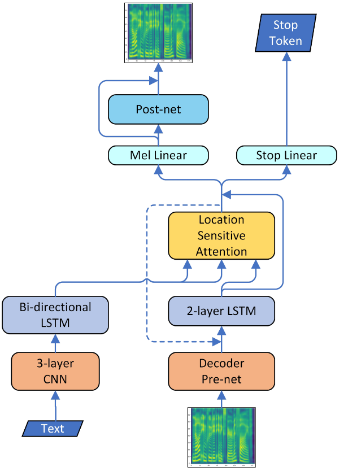


Image taken from `https://arxiv.org/pdf/1809.08895.pdf`

Transformer for TTS

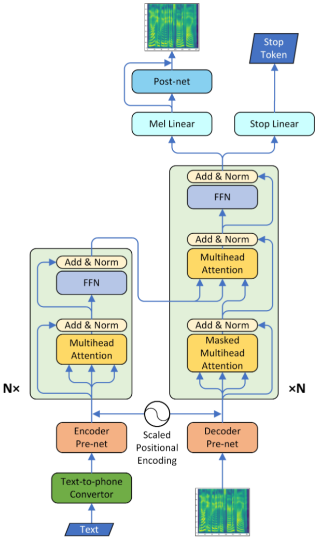
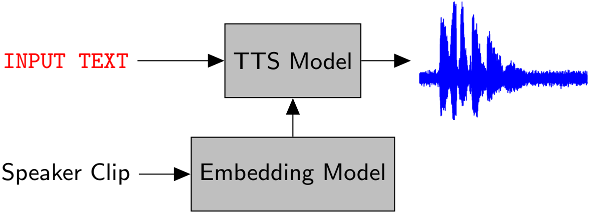


Image taken from `https://arxiv.org/pdf/1809.08895.pdf`

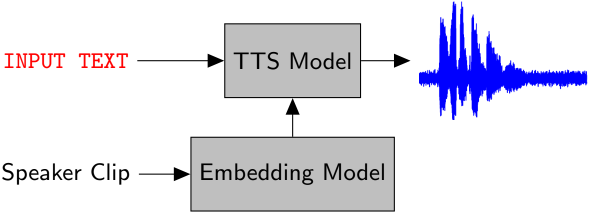
Speaker Embeddings for Zero-Shot Speaker Generalization in Multi-Speaker TTS

- Adaptation of a multi-speaker TTS system with voice snippets.



Speaker Embeddings for Zero-Shot Speaker Generalization in Multi-Speaker TTS

- Adaptation of a multi-speaker TTS system with voice snippets.



- **Examples (Unseen speakers):**
  - Speaker1 Clip, Speaker1 UnseenPhrase
  - Speaker2 Clip, Speaker2 UnseenPhrase
  - Speaker3 Clip, Speaker3 UnseenPhrase
- **Goals:** Improving speaker generalization, Increasing clip invariance through better speaker embeddings
  - Speaker1 Clip, Speaker1 UnseenPhrase

ECAPA-TDNN for speaker embeddings

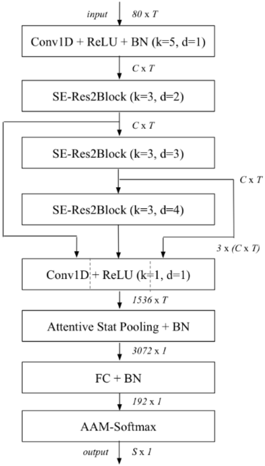


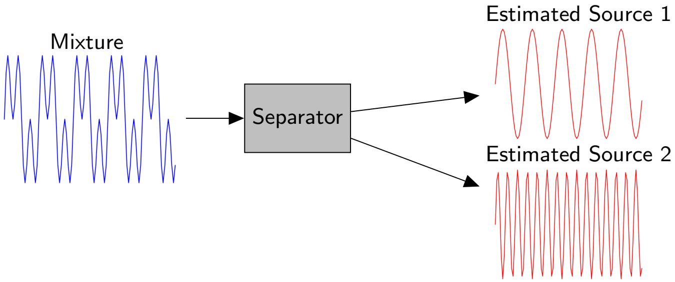
image taken from the original paper
`https://arxiv.org/pdf/2005.07143.pdf`

- Model pretrained on speaker-id works well for speaker embeddings.
- Pretrained model on SpeechBrain Huggingface `https://huggingface.co/speechbrain/spkrec-ecapa-voxceleb`
- There's also the X-Vector model, which is a smaller. `https://huggingface.co/speechbrain/spkrec-xvect-voxceleb`

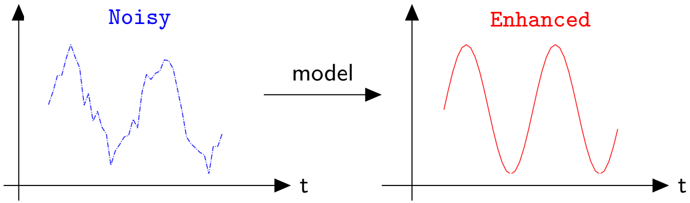
Table of Contents

Table of Contents

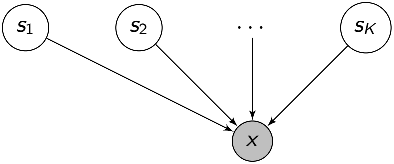
Source Separation



Speech Enhancement



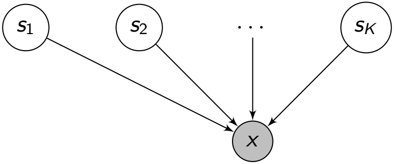
Source Separation



- The observation x is dependent on latent factors $s_1, s_2, \ldots, s_K$.
  - Technical definition:

$$s_1 \sim p(s_1) \;\ldots\; s_K \sim p(s_K)$$
$$x \sim p(x|s_1, \ldots, s_K)$$

Source Separation



- The observation x is dependent on latent factors $s_1, s_2, \ldots, s_K$.
  - ▶ Technical definition:

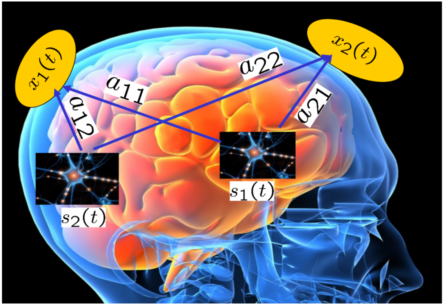
$$s_1 \sim p(s_1) \;\ldots\; s_K \sim p(s_K)$$
$$x \sim p(x|s_1, \ldots, s_K)$$

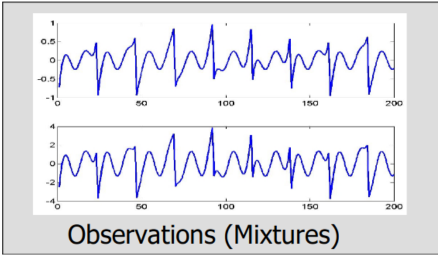
- Additive separation model:

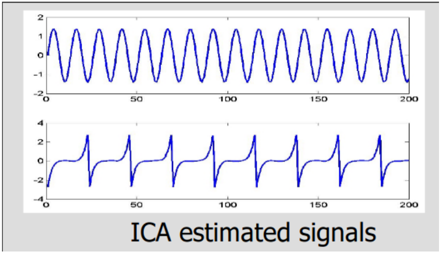
$$x(t) = \sum_{k=1}^{K} a_k s_k(t)$$

  - ▶ This is a very general formulation and captures several diffent models / algorithms.
    E.g. PCA, ICA, Factor Analysis, Mixture models, NMF, HMMs, Linear Dynamical Systems (Kalman filters)

Source Separation Application

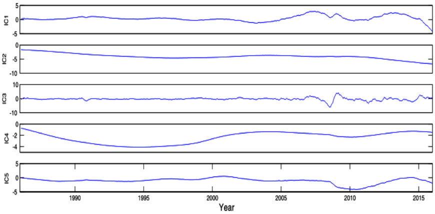






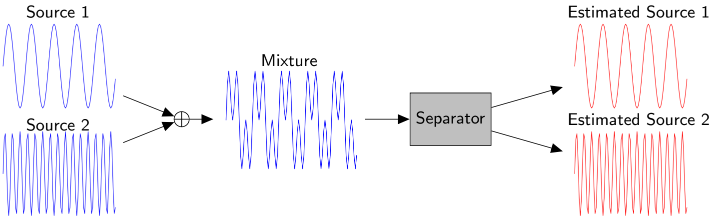
[images taken from https://www.cs.cmu.edu/~bapoczos/other_presentations/ICA_26_10_2009.pdf]

Source Separation for Financial Data



- In the paper Factor analysis of financial time series using EEMD-ICA based approach the authors decompose oil prices using an ICA variant.
- They claim:
  - IC1 is correlated to USD.
  - IC2 is correlated to oil suppy and demand.
  - IC3 is correlated to political and extreme events.
  - IC4 reflects cyclical nature of oil prices.
  - IC5 is correlated with stock, gold markets.

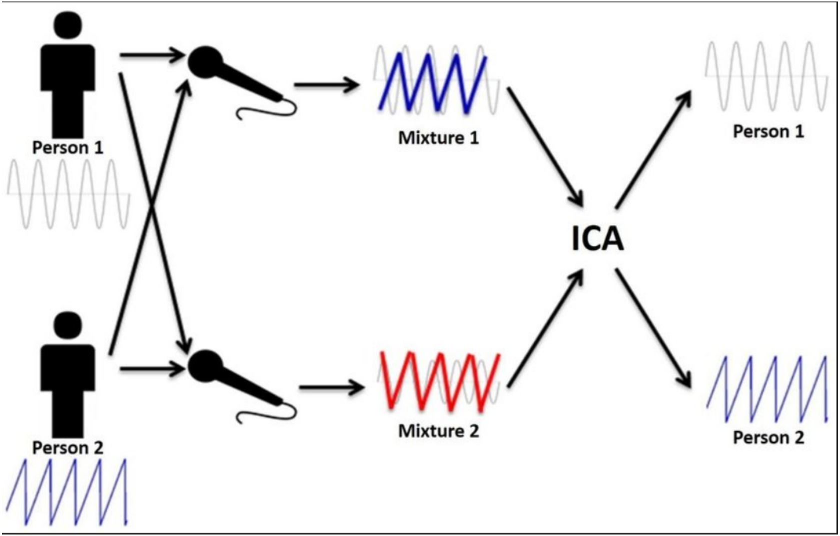
Single-Microphone Source Separation Problem



- **Goal:** To recover the original sources from the observed mixture
- **Applications:** Music production, hearing devices, meeting analysis, editing software, and more...
- **Some of my contributions**
  - Hierarchical tensor factorizations
  - Globally optimal unsupervised source separation with FHMM.
  - Neural network analogs to matrix factorization (best paper award)
  - GANs in source separation
  - **SepFormer**, a self-attention based source separation architecture and obtain state-of-the-art results on multiple datasets.
  - **REAL-M** dataset and evaluation framework

Table of Contents

Independent Component Analysis



Independent Component Analysis

$$x_{1:T} = \sum_{k=1}^{K} a_k s_{1:T}^k$$

$$\underbrace{\begin{pmatrix} \dots & x_{1:T}^1 & \dots \\ \dots & x_{1:T}^2 & \dots \\ \dots & \vdots & \dots \\ \dots & x_{1:T}^N & \dots \end{pmatrix}}_{Mixtures} = \underbrace{\begin{pmatrix} a_{1,1} & \dots & a_{1,K} \\ a_{2,1} & \dots & a_{2,K} \\ \dots & \dots & \dots \\ a_{N,1} & \dots & a_{N,K} \end{pmatrix}}_{Mixing\ Matrix} \underbrace{\begin{pmatrix} \dots & s_{1:T}^1 & \dots \\ \dots & s_{1:T}^2 & \dots \\ \dots & \vdots & \dots \\ \dots & s_{1:T}^K & \dots \end{pmatrix}}_{Sources}$$

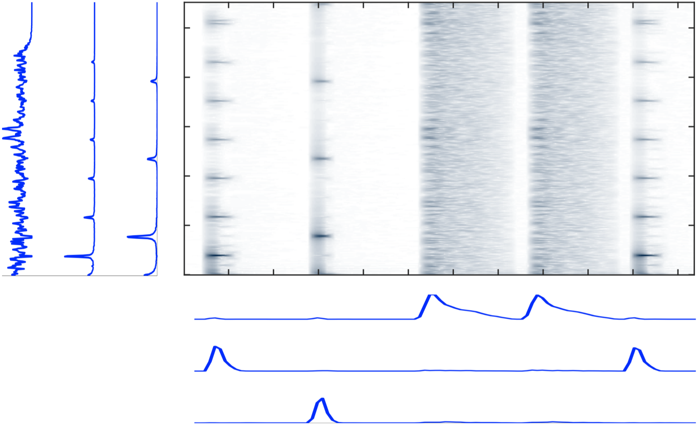
- Unsupervised!
- We try to estimate statistically independent components $s^{1:K}$.
- 2×2 case:

$$\begin{pmatrix} \dots & x_{1:T}^1 & \dots \\ \dots & x_{1:T}^2 & \dots \end{pmatrix} = \begin{pmatrix} a_{1,1} & a_{1,2} \\ a_{2,1} & a_{2,2} \end{pmatrix} \begin{pmatrix} \dots & s_{1:T}^1 & \dots \\ \dots & s_{1:T}^2 & \dots \end{pmatrix}$$

- Works in time domain, and requires $N \geq K$.

Non-Negative Matrix Factorization

[Lee, Seung 1999][Smaragdis 2003, Non-Negative Matrix Factorization for Polyphonic Music Transcription]



Popular NMF model: $X = WH$

$$\min_{W,H} \|X - WH\|, s.t.\ W \geq 0,\ H \geq 0$$

Early deep learning approaches

[Huang et al. 2014, Deep Learning for Monoaural Speech Separation], figure taken from the paper.

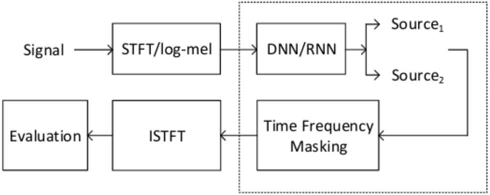


Fig. 1: Proposed framework

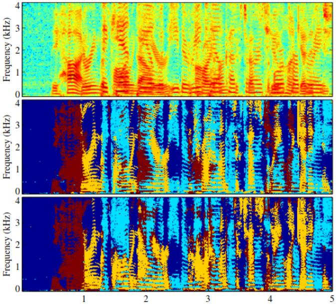
- Works on magnitude STFTs.
- Uses the mixture phase to reconstruct

Deep Clustering

[Hershey et al. 2015, Deep Clustering], The idea is the find an embedding for the affinity matrix.

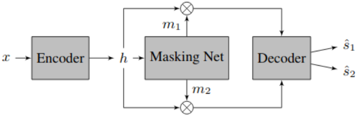
$$\min_{\theta} \| YY^\top - f_\theta(x) f_\theta(x)^\top \| \tag{1}$$

They learn an embedding for each time-frequency bin, and minimize this affinity based loss. In test time, they cluster the embeddings.



End-to-End training: Masking based architecture

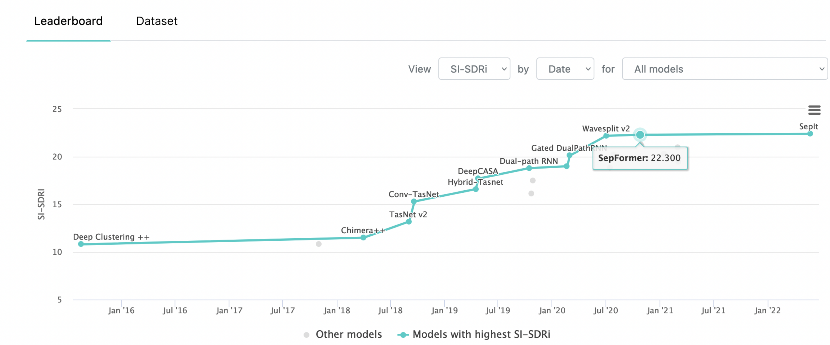
- The masking based architecture [Luo, Mesgarani 2018, Convtasnet],



- **Encoder:** A time-transformation representation is calculated by passing via the encoder. Special case: STFT
- **Masking Network:** A masking network estimates an element-wise mask m_i for each source.
- **Decoder:** For each source i, we reconstruct the estimated source by passing the filtered representation $h * m_i$ through the decoder.

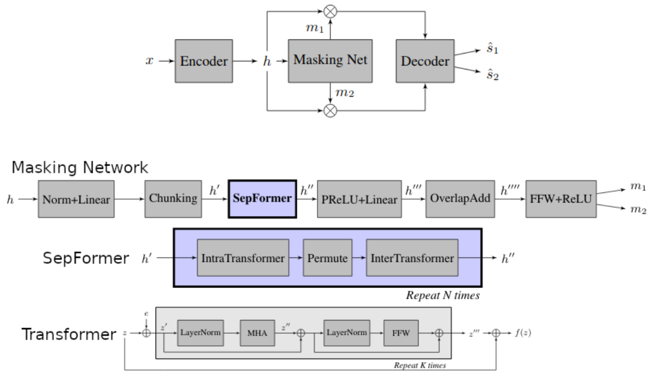
Table of Contents

WSJ0-2Mix Leaderboard

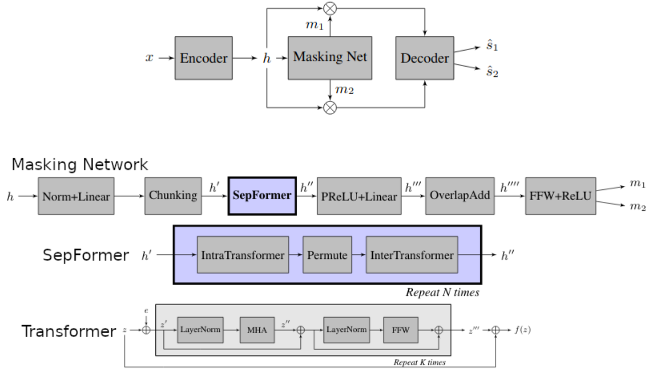


- Taken from `https://paperswithcode.com/sota/speech-separation-on-wsj0-2mix` on October 2022. SepFormer stayed state of the art on WSJ0-2Mix from October 2020-September 2022.

The SepFormer Architecture



The SepFormer Architecture

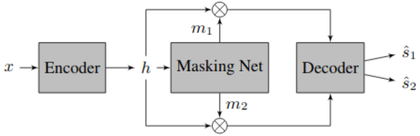


We train this architecture with permutation invariant SI-SNR.

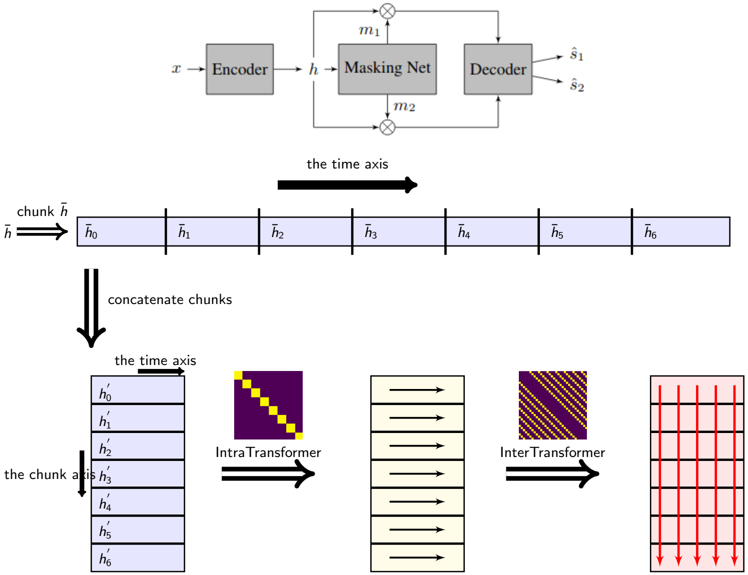
$$s_{\text{target}} := \frac{\widehat{s}^{\top} s}{\|s\|^2} s, \quad e_{\text{noise}} := \widehat{s} - s_{\text{target}}, \quad \text{SI-SNR} := 10 \log_{10} \left(\frac{\|s_{\text{target}}\|^2}{\|e_{\text{noise}}\|^2} \right)$$

$$\text{PIT-SISNR} = \sum_k \min_{k' \in \mathcal{P}} 10 \log_{10} \left(\frac{\|s^k_{\text{target}}\|^2}{\|\widehat{s}^{k'} - s^k_{\text{target}}\|^2} \right)$$

SepFormer Architecture



SepFormer Architecture



Best Results on Mixtures of 2 speakers (WSJ0-2Mix)

Model	SI-SNRi	SDRi	# Param	Stride
Tasnet	10.8	11.1	n.a	20
SignPredictionNet	15.3	15.6	55.2M	8
ConvTasnet	15.3	15.6	5.1M	10
Two-Step CTN	16.1	n.a.	8.6M	10
DeepCASA	17.7	18.0	12.8M	1
FurcaNeXt	n.a.	18.4	51.4M	n.a.
DualPathRNN	18.8	19.0	2.6M	1
sudo rm -rf	18.9	n.a.	2.6M	10
VSUNOS	20.1	20.4	7.5M	2
DPTNet	20.2	20.6	2.6M	1
Wavesplit	22.2	22.3	29M	1
SepFormer	**22.3**	**22.4**	26M	8

$$SNR \propto 10 \log \left(\frac{\text{Ener. Signal}}{\text{Ener. Noise}} \right)$$

Best Results on Mixtures of 3 Speakers (WSJ0-3Mix)

Model	SI-SNRi	SDRi	# Param
ConvTasnet	12.7	13.1	5.1M
DualPathRNN	14.7	n.a	2.6M
VSUNOS	16.9	n.a	7.5M
Wavesplit	17.8	18.1	29M
Sepformer	**19.5**	**19.7**	26M

Example Results on Test Set:

`Click for Mixture`

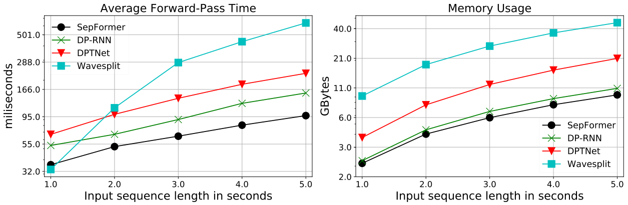
`Click for Estimated Source1`

`Click for Estimated Source2`

`Click for Estimated Source3`

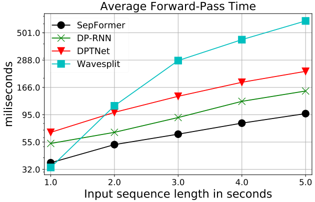
Speed/Memory Comparison with Other Methods

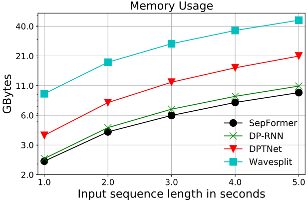
Speed and Memory Comparison on Forward Pass:



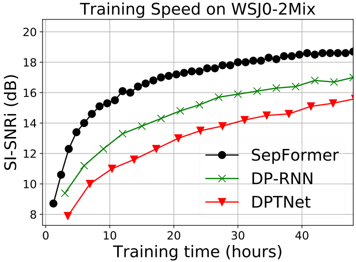
Speed/Memory Comparison with Other Methods

Speed and Memory Comparison on Forward Pass:





Training Curve Comparison:



Environmental Corruption

We try our model with environmental noise / reverberation.

Best results on the WHAM dataset (noise).

Model	**SI-SNRi**	**SDRi**
ConvTasnet	12.7	-
Learnable fbank	12.9	-
Wavesplit	16.0	16.5
Sepformer	**16.4**	**16.7**

Best results on the WHAMR (noise + reverb) dataset.

Model	**SI-SNRi**	**SDRi**
ConvTasnet	8.3	-
BiLSTM Tasnet	9.2	-
Wavesplit	13.2	12.2
Sepformer	**14.0**	**13.0**

Cross-Dataset Experiment

We test our model trained on WSJ0-2Mix on LibriMix.

Model	**SI-SNRi**	**SDRi**
ConvTasnet	14.7	-
Sepformer trained on WSJ0-2Mix	17.0	17.5
Wavesplit	20.5	20.7
Sepformer	20.2	20.5
Sepformer + FT	**20.6**	**20.8**

We test our model trained on WSJ0-3Mix on LibriMix.

Model	**SI-SNRi**	**SDRi**
ConvTasnet	10.4	-
Sepformer trained on WSJ0-3Mix	15.0	15.6
Wavesplit	17.5	18.0
Sepformer	18.2	18.6
Sepformer + FT	**18.7**	**19.0**

Note: We release our pretrained models, training scripts on SpeechBrain!

Synthetic vs Real Life Mixture

Synthetic: WSJ0-2Mix test set

`Click for Mixture`

`Click for Estimated Source1`

`Click for Estimated Source2`

Real-life: One mic, two people speaking, reverberant environment

`Click for Mixture`

`Click for Estimated Source1`

`Click for Estimated Source2`

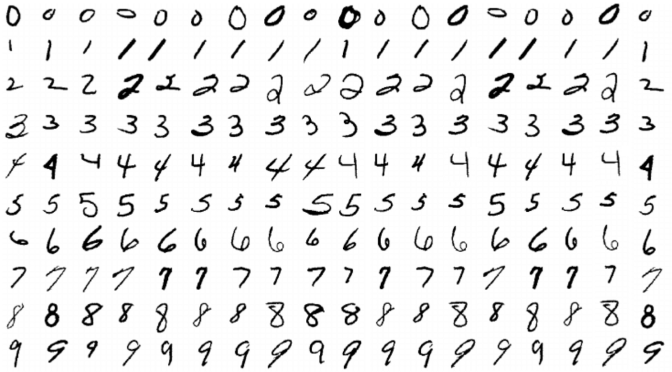
`Click for Mixture`

`Click Estimated Source 1`

`Click Estimated Source 2`

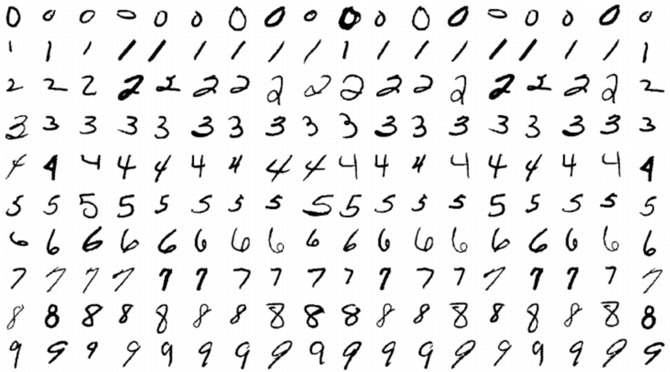
Table of Contents

Real-life machine learning



- A lot of machine learning focuses on improving on benchmarks
- This is good, but it's likely that there is a reality-gap.

Reality GAP on MNIST



- Small, fixed size images
- Perfectly aligned images,
- Uniform backgrounds
- **We need at least an evaluation set for evaluating models on real-life data.**

Closing the reality gap

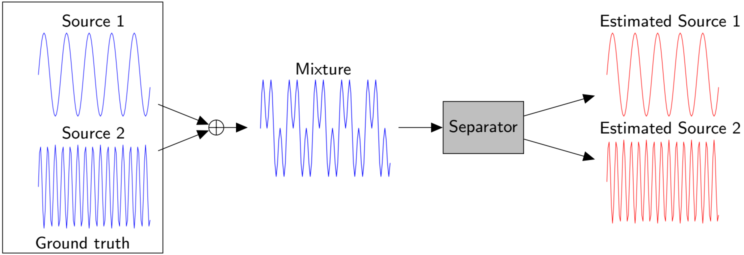
- We need evaluation sets that represents the challanges of real-life so that researchers can more meaningfully benchmark their performance.
- We can then design data augmentations, and models to improve performance on real-life data.

Closing the reality gap

- We need evaluation sets that represents the challanges of real-life so that researchers can more meaningfully benchmark their performance.
- We can then design data augmentations, and models to improve performance on real-life data.
- An important hurdle: Ground truth data.

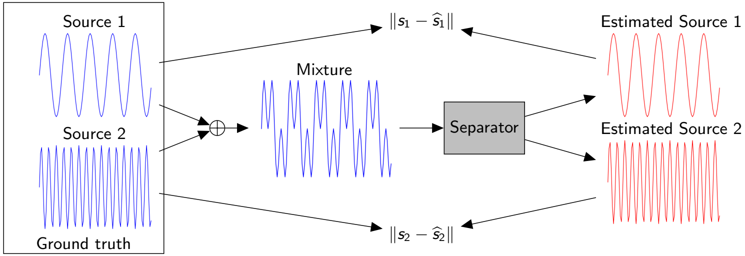
Lack of ground truth in real-life separation

- Synthetic source separation datasets



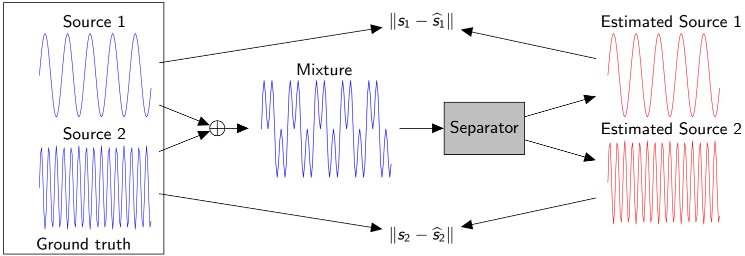
Lack of ground truth in real-life separation

- Synthetic source separation datasets

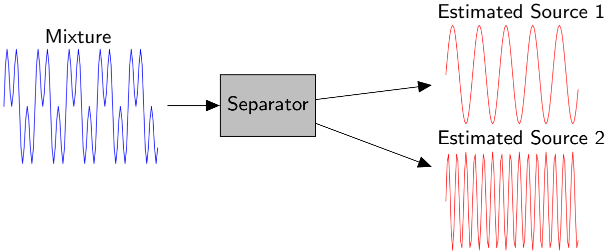


Lack of ground truth in real-life separation

- Synthetic source separation datasets

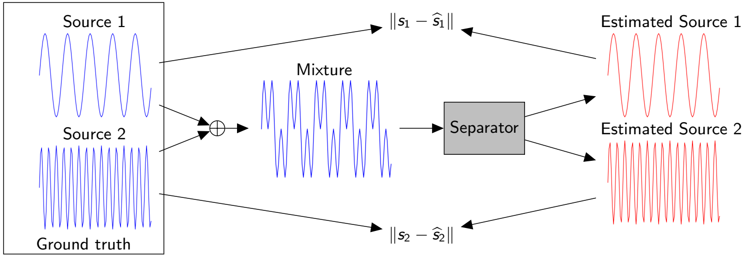


- Real-life source separation

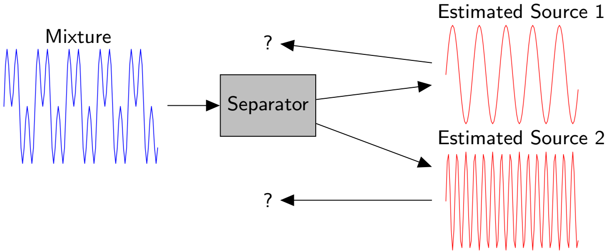


Lack of ground truth in real-life separation

- Synthetic source separation datasets



- Real-life source separation

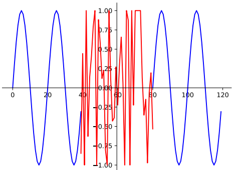


Tackling the lack of ground truth

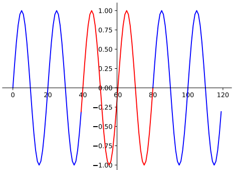
- In real-life datasets we often do not have the ground truth information.
- Examples:
  - Image in-painting
  - Speech enhancement and separation
  - Image super-resolution
  - Evaluating the performance of chatbots
  - Website design optimization to maximize user retention
- The lack of ground truth prevents evaluating estimation quality on real-life data.

Tackling the lack of ground truth

- In real-life datasets we often do not have the ground truth information.
- Examples:
  - Image in-painting
  - Speech enhancement and separation
  - Image super-resolution
  - Evaluating the performance of chatbots
  - Website design optimization to maximize user retention
- The lack of ground truth prevents evaluating estimation quality on real-life data.
- We can however estimate the performance!
- We can train a model to estimate the performance.



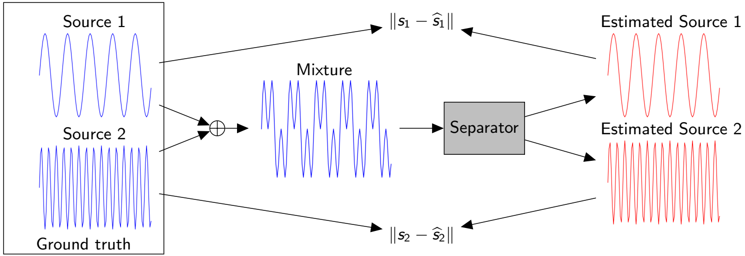
WORSE



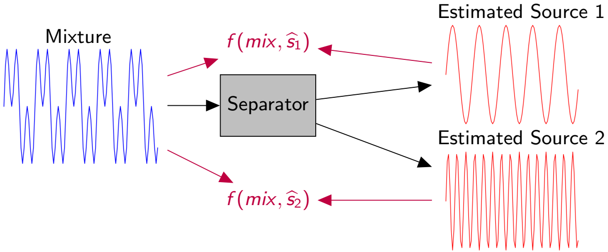
GOOD

Tackling the lack of ground truth

- Synthetic source separation datasets



- Real-life source separation



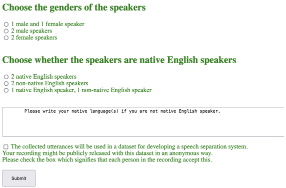
REAL-M: Towards Speech Separation on Real Mixtures

- **Goal: Systematic Evaluation of Speech Separation Models on Real-Life Speech Mixtures**.
- **Contributions:**
  - We propose a dataset for **real-life speech separation**. The dataset is **crowdsourced**, hence **scalable and diverse** in acoustic conditions, recording hardware, speakers.
  - We show that **blind SI-SNR estimation** is a feasible way to evaluate real-life speech separation.
  - Therefore, this opens up a scalable methodology for large-scale real-life source separation evaluation.
  - The 5th most viewed poster in ICASSP 2022! (out of 1900 posters)

Table of Contents

Data collection

- We crowdsource a dataset to construct an evaluation of audio mixtures in real-life. `Click to hear examples`.



Choose the genders of the speakers

- ○ 1 male and 1 female speaker
- ○ 2 male speakers
- ○ 2 female speakers

Choose whether the speakers are native English speakers

- ○ 2 native English speakers
- ○ 2 non-native English speakers
- ○ 1 native English speaker, 1 non-native English speaker

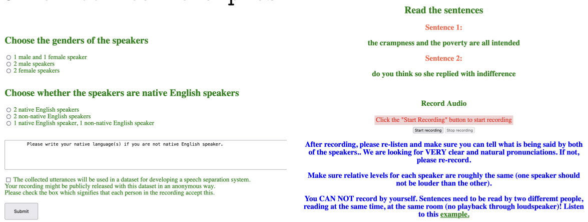
Please write your native language(s) if you are not native English speaker.

☐ The collected utterances will be used in a dataset for developing a speech separation system.
Your recording might be publicly released with this dataset in an anonymous way.
Please check the box which signifies that each person in the recording accept this.

Submit

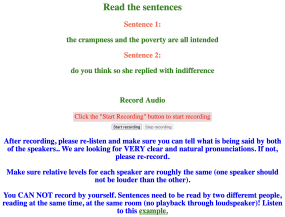
Data collection

- We crowdsource a dataset to construct an evaluation of audio mixtures in real-life. `Click to hear examples`.



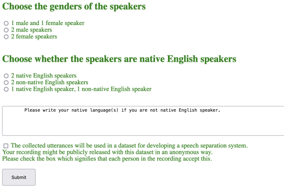
Data collection

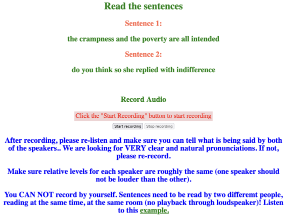
- We crowdsource a dataset to construct an evaluation of audio mixtures in real-life. `Click to hear examples`.

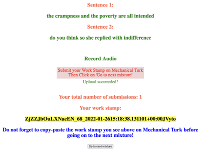


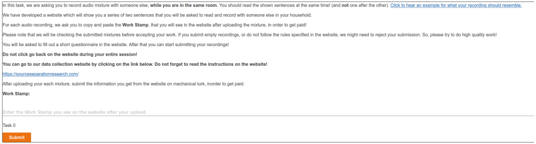
Data collection

- We crowdsource a dataset to construct an evaluation of audio mixtures in real-life. `Click to hear examples`.









- Contributors are asked simultaneously read the shown sentences.
- This gives a way to collect real-life speech mixtures in a scalable way. We interface our platform with Mechanical Turk.
- We collected 3 hours of speech, from 50 unique speakers, with various native (e.g. US, UK) and non-native (e.g. French, Italian, Persian, Indian, African) accents, in various conditions, with various recording equipment.

Table of Contents

SI-SNR Estimator Training Pipeline

- We construct a performance estimator $f(.)$ such that,

$$f(x, \widehat{s}) \approx \textbf{SI-SNR}(s, \widehat{s}),$$

where $f(.)$ is a neural network, x is the mixture, $\widehat{s}$ is the source estimate, s is the true source.

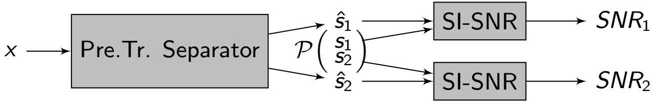
SI-SNR Estimator Training Pipeline

- We construct a performance estimator $f(.)$ such that,

$$f(x, \widehat{s}) \approx \textbf{SI-SNR}(s, \widehat{s}),$$

  where $f(.)$ is a neural network, x is the mixture, $\widehat{s}$ is the source estimate, s is the true source.

- We train the estimator on synthetic mixtures on which ground truth information is available. (Also a lot of data!)



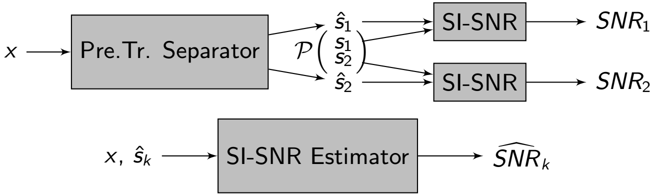
SI-SNR Estimator Training Pipeline

- We construct a performance estimator $f(.)$ such that,

$$f(x, \widehat{s}) \approx \textbf{SI-SNR}(s, \widehat{s}),$$

  where $f(.)$ is a neural network, x is the mixture, $\widehat{s}$ is the source estimate, s is the true source.

- We train the estimator on synthetic mixtures on which ground truth information is available. (Also a lot of data!)



$$\mathcal{L} \;=\; \|SNR_1 - \widehat{SNR}_1\| + \|SNR_2 - \widehat{SNR}_2\|.$$

- SI-SNR Estimator is a 5-layer convolutional NNet in the time domain.

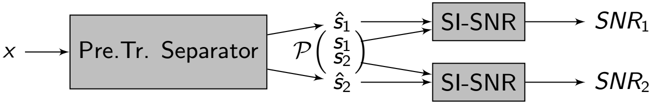
SI-SNR Estimator Training Pipeline

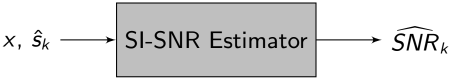
- We construct a performance estimator $f(.)$ such that,

$$f(x, \widehat{s}) \approx \textbf{SI-SNR}(s, \widehat{s}),$$

  where $f(.)$ is a neural network, x is the mixture, $\widehat{s}$ is the source estimate, s is the true source.

- We train the estimator on synthetic mixtures on which ground truth information is available. (Also a lot of data!)





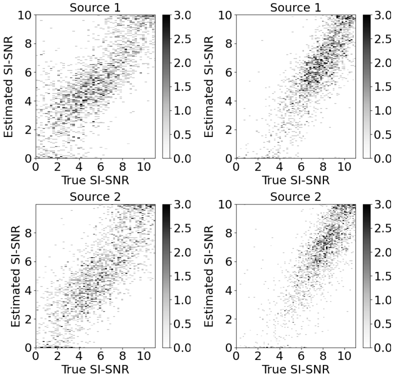
$$\mathcal{L} \;=\; \|SNR_1 - \widehat{SNR}_1\| + \|SNR_2 - \widehat{SNR}_2\|.$$

- SI-SNR Estimator is a 5-layer convolutional NNet in the time domain.
- **Important Question:** Is this estimator going to work well (generalize to) real-mixtures?

Table of Contents

Evaluating the SI-SNR Estimator

- We first evaluate the SI-SNR Estimator on synthetic data.



- Evaluating on (left) LibriMix, (right) WHAMR!
- Both scatter plots correspond to Pearson correlation coefficient of 0.8.

Improving the SI-SNR Estimator

- We train with multiple separators.
- We observe improvement in the pearson correlation coefficient when we train with multiple separators.

	SI-SNR-Estimator 1 *(single)*		SI-SNR-Estimator 2 *(pool)*	
Model	**LibriMix**	**WHAMR!**	**LibriMix**	**WHAMR!**
SF	0.80	0.81	0.82	0.87
DPRNN	0.80	0.80	0.83	0.84
CTN	0.81	0.79	0.85	0.86

Improving the SI-SNR Estimator

- We train with multiple separators.
- We observe improvement in the pearson correlation coefficient when we train with multiple separators.

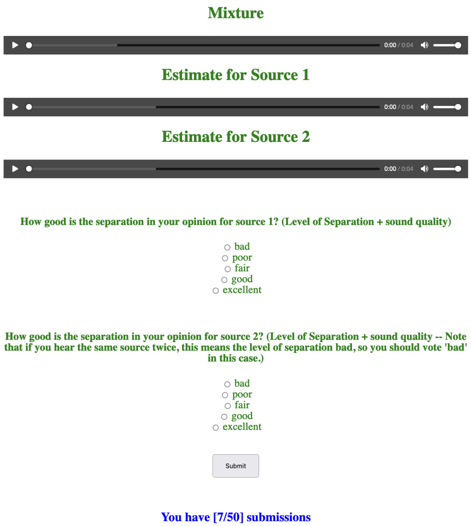
	SI-SNR-Estimator 1 *(single)*		SI-SNR-Estimator 2 *(pool)*	
Model	**LibriMix**	**WHAMR!**	**LibriMix**	**WHAMR!**
SF	0.80	0.81	0.82	0.87
DPRNN	0.80	0.80	0.83	0.84
CTN	0.81	0.79	0.85	0.86

- **Important Question:** Is this estimator going to work well (generalize to) real-mixtures?

Table of Contents

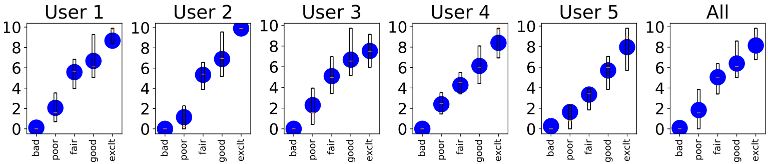
Evaluating the SI-SNR Estimator on REAL-M

- We validate the SI-SNR estimator with a user study on real-life data.
- We presented 50 random mixtures and the separation results to 5 users.
- We asked the users to rate the presented separation result between 1-5.



Results of User Study

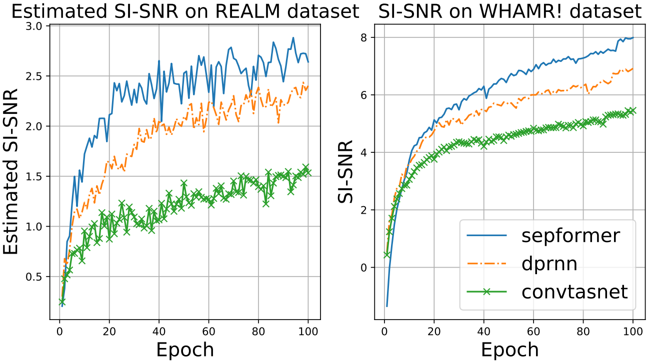
- Results of the user study suggest that on average the opinion scores correlate-well with SNR estimates.



- Y-axes show the estimated-SNR, X-axes show the user rating.

Further evaluation of SI-SNR Estimator

- The performance rankings of models on synthetic data holds true for REAL-M as well.
- We also observe that with training epochs performance on REAL-M dataset improves.



Conclusions

- With the REAL-M framework, we are moving towards working with real-mixtures. It provides,
  - Scalable data collections
  - Variability
  - Blind Performance estimation

Conclusions

- With the REAL-M framework, we are moving towards working with real-mixtures. It provides,
  - Scalable data collections
  - Variability
  - Blind Performance estimation
- The performance on real-life data is far worse than synthetic benchmarks.

Separator	$\widehat{\textbf{SNR Synth}}$	$\widehat{\textbf{SNR Real}}$
SepFormer	8.40	2.88
DPRNN	7.04	2.43
CTN	5.49	1.59

Conclusions

- With the REAL-M framework, we are moving towards working with real-mixtures. It provides,
  - Scalable data collections
  - Variability
  - Blind Performance estimation
- The performance on real-life data is far worse than synthetic benchmarks.

Separator	$\widehat{\textbf{SNR}}$ **Synth**	$\widehat{\textbf{SNR}}$ **Real**
SepFormer	8.40	2.88
DPRNN	7.04	2.43
CTN	5.49	1.59

- **Next steps:**
  - Casual talking, Meeting settings
  - Scaling up

Conclusions

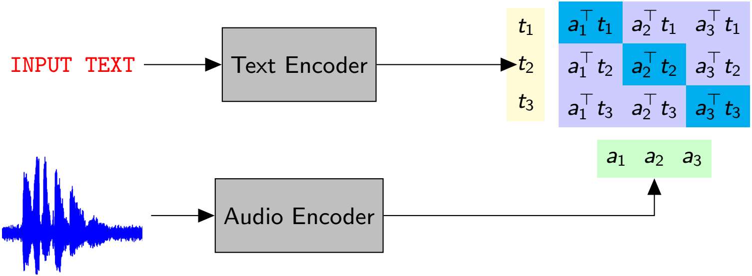
- With the REAL-M framework, we are moving towards working with real-mixtures. It provides,
  - Scalable data collections
  - Variability
  - Blind Performance estimation
- The performance on real-life data is far worse than synthetic benchmarks.

Separator	**$\widehat{\text{SNR}}$ Synth**	**$\widehat{\text{SNR}}$ Real**
SepFormer	8.40	2.88
DPRNN	7.04	2.43
CTN	5.49	1.59

- **Next steps:**
  - Casual talking, Meeting settings
  - Scaling up
  - Improving the generalization (Data augmentations, Using the performance estimators, Using pretrained models, ...)

Table of Contents

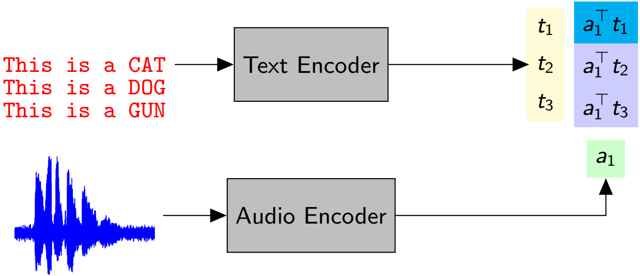
Cross-Modal Representation Learning



- CLAP: Contrastive Language-Audio Pretraining
  - We maximize $a_i^\top t_j$ for $i = j$, and minimize for $i \neq j$.
  - This enables text-based audio retrieval, zero-shot classification.

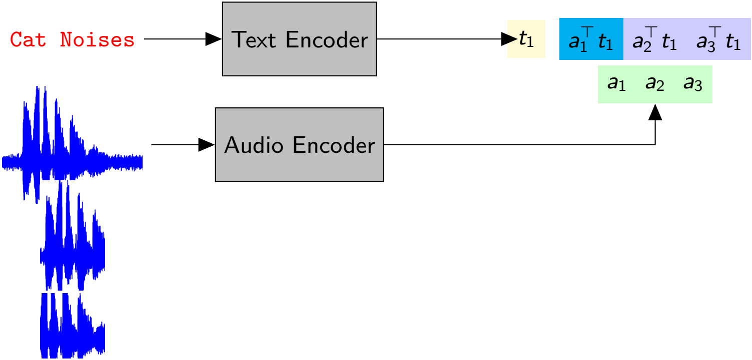
Cross-Modal Representation Learning

- Zero-shot evaluation

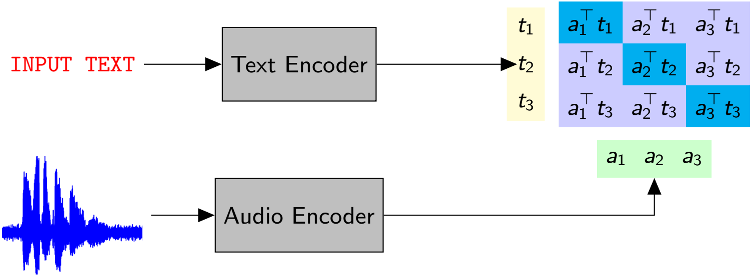


Cross-Modal Representation Learning

- Audio Retreival

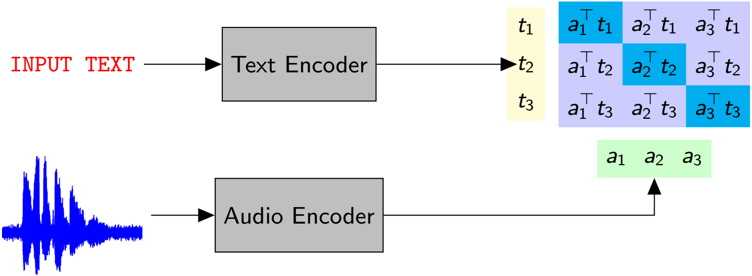


Cross-Modal Representation Learning



- CLAP
  - Training this model requires large number of paired data.

Cross-Modal Representation Learning



- CLAP
  - Training this model requires large number of paired data.
  - **WASPAA 2023 paper:** We published a paper where we improve the zero-shot classification performance using unpaired text and audio.

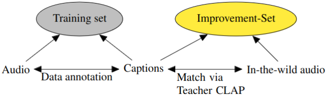
CLAP training objective

- Audio and text encoders $L_a = f_a(X_a)$, $L_t = f_t(X_t)$.
- The joint space: $a = MLP_a(L_a)$, $t = MLP_t(L_t)$
- We maximize the diagonal here $C = ta^\top$, through this loss function:

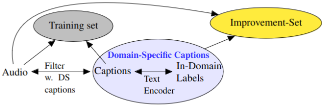
$$\mathcal{L}(C) = \frac{1}{2}\sum_{i=1}^{N}\Big(\log(\mathbf{Softmax}_t(C/\tau)_{i,i}) + \log(\mathbf{Softmax}_a(C/\tau)_{i,i})\Big),$$

UICLAP

- We bootstrap CLAP by self-training. We explore different strategies for bootstrapping.



Model	Zero-Shot Evaluation Set ESC-50	UrbanSound8K	TUT17
CLAP teacher	81.9 ± 0.9	74.8 ± 1.2	29.8 ± 1.3
SL	83.1 ± 1.2	73.9 ± 2.6	30.1 ± 2.1
DU	82.4 ± 1.4	73.9 ± 0.2	31.5 ± 1.0
DU+SL	83.0 ± 0.5	74.9 ± 1.4	29.9 ± 1.9
DS	78.8 ± 0.5	73.2 ± 1.1	29.8 ± 2.6
DS+SL	83.5 ± 0.6	75.5 ± 1.4	31.8 ± 2.6
ADS	84.2 ± 0.5	74.2 ± 2.1	32.5 ± 1.0
ADS+SL	**85.1** ± 0.7	**77.4** ± 0.6	**36.0** ± 1.8



Model	Zero-Shot Evaluation Set ESC-50	UrbanSound8K	TUT17
CLAP teacher (full-dataset)	81.9 ± 0.9	74.8 ± 1.2	29.8 ± 1.3
CLAP teacher (subset)	74.2 ± 1.3	73.5 ± 2.0	30.9 ± 1.6
DU + SL	78.9 ± 0.3	73.7 ± 1.3	28.8 ± 1.1
ADS + SL	**81.3** ± 1.0	**74.5** ± 0.7	**31.3** ± 0.5

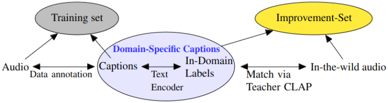


Table of Contents

Table of Contents

Neural Network Explanation

- *Why does this particular input lead to that particular output?*

Neural Network Explanation

- *Why does this particular input lead to that particular output?*

Neural Network Explanation

- *Why does this particular input lead to that particular output?*

Recording, Classified as DOG

Neural Network Explanation

- *Why does this particular input lead to that particular output?*

Recording, Classified as DOG

Interpretation

- **Posthoc Explanation vs Explainable Models:** Posthoc Explanation produces explanations for already trained models. Explainable are by design so.

Listen-to-Interpret

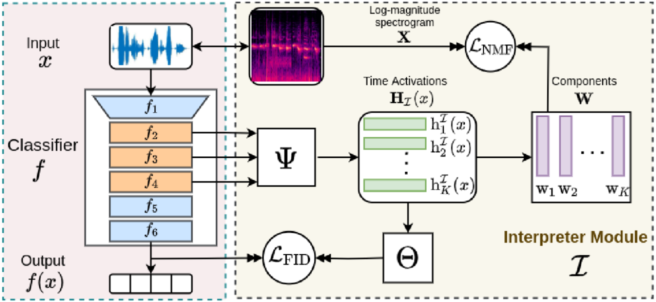
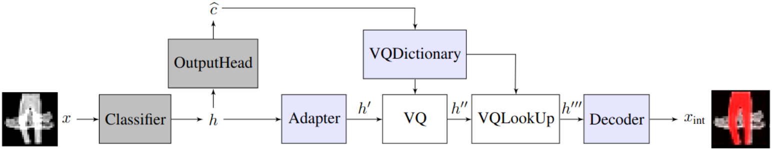


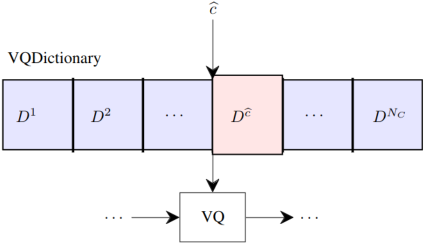
Image taken from https://arxiv.org/pdf/2202.11479.pdf

Posthoc Interpretation via Quantization

- We have developed a method that learns "high-level" concepts for each class in form of latent VQ dictionary, and then reconstructs the input using this VQ dictionary conditioned on the class information.

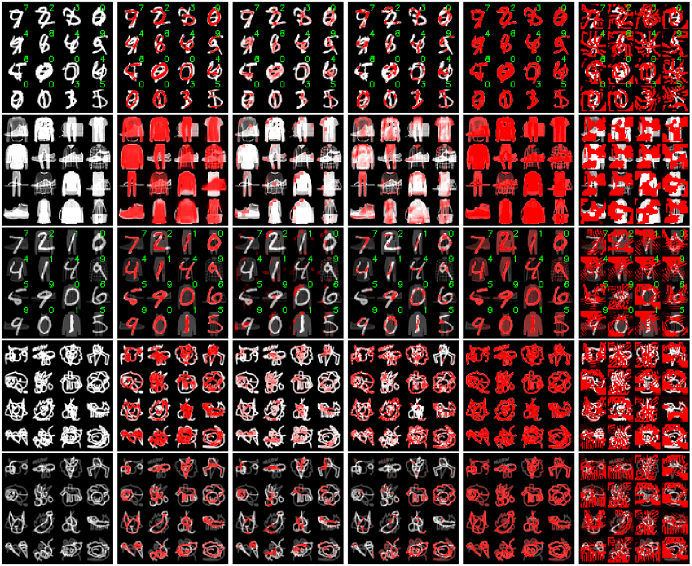
Posthoc Interpretation via Quantization





Above shows the inference time. In training, we only use images with single classes. NOT mixtures.

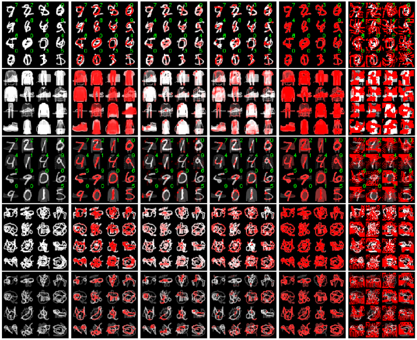
Qualitative Results on Images



Left-to-Right: Input, PIQ (ours), VIBI, L2I, LIME, FLINT

Mean-Opinion-Scores on Images

Dataset	Method	MOS (↑)
MNIST B1 (Case 1)	PIQ (ours)	**4.04 ± 0.48**
	VIBI	1.77 ± 0.68
	L2I	2.4 ± 0.66
	FLINT	1 ± 0
	LIME	2 ± 1.34
MNIST B2 (Case 1)	PIQ (ours)	**3.95 ± 0.72**
	VIBI	1.86 ± 0.71
	L2I	1.86 ± 0.56
	FLINT	1.04 ± 0.21
	LIME	2.13 ± 1.21
FMNIST Mix (Case 2)	PIQ (ours)	**4.87 ± 0.50**
	VIBI	1.37 ± 0.50
	L2I	3.18± 0.91
	FLINT	1.12± 0.50
	LIME	1.37 ± 0.89
MNIST+FMN (Case 3)	PIQ (ours)	**4.78 ± 0.43**
	VIBI	1.14 ± 0.47
	L2I	2.18 ± 0.96
	FLINT	1.09 ± 0.47
	LIME	3.23 ± 0.72
QuickDraw1 (Case4-i)	PIQ (ours)	**2.6 ± 1.67**
	LIME	2.35 ± 1.46
QuickDraw2 (Case4-ii)	PIQ (ours)	**3.55 ± 1.0**
	LIME	3 ± 1,38



Quantitative Results on Images

Dataset	MNIST			FMNIST		
Metric	Fidelity-In (↑)	Faithfulness (↑)	FID (↓)	Fidelity-In (↑)	Faithfulness (↑)	FID (↓)
PIQ (ours)	**98.03 ± 0.05**	**0.588 ± 0.00021**	**0.029 ± 0.0004**	**81.3 ± 0.2**	**0.773 ± 0.004**	**0.030 ± 0.0004**
VIBI	73.90 ± 16.08	0.369 ± 0.002	0.710 ± 0.962	42.4 ± 17.8	0.578 ± 0.073	0.395 ± 0.104
L2I	96.56 ± 2.66	0.453 ± 0.002	0.160 ± 0.010	68.3 ± 1.5	0.343 ± 0.011	0.188 ± 0.011
FLINT	10.9	0.361	0.677	15.37	-0.097	0.482

Dataset	Quickdraw		
Metric	Fidelity-In (↑)	Faithfulness (↑)	FID (↓)
PIQ (ours)	**60.89 ± 0.60**	**0.675 ± 0.005**	**0.034 ± 0.0001**
VIBI	26.36 ± 3.01	0.341 ± 0.031	0.388 ± 0.032
L2I	25.97 ± 0.82	0.340 ± 0.031	0.397 ± 0.020
FLINT	15.62	-0.057	0.672

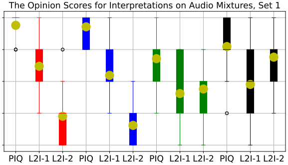
- Input Fidelity

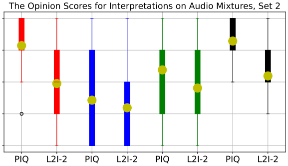
$$\text{FID-I} = \frac{1}{N}\sum_{n=1}^{N}\left[\arg\max_c f_c(x_n) = \arg\max_c f_c(x_{\text{int},n})\right],$$

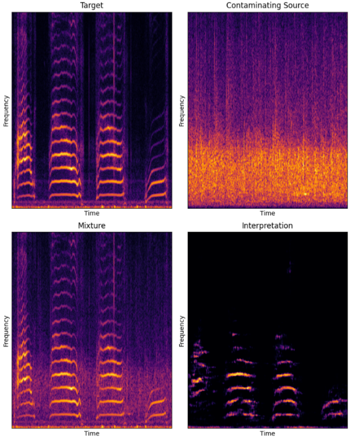
- Faithfulness

$$\text{Faithfulness} = f_{\hat{c}}(x) - f_{\hat{c}}(x - x_{\text{int}}),$$

Mean-Opinion Scores on Audio







Click for More Example Results

Ongoing projects in Interpretability

- Exploring Understandability / Faithfulness Tradeoffs
- Interpretable Detection of Fake vs Real Audio
- Listenable Recommendation Systems
- Interpretable Baby Cry Analysis

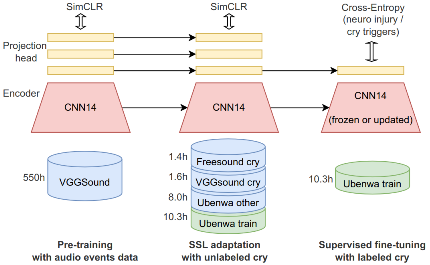
Table of Contents

ML for Infant Cry Analysis

- Collaboration with UbenwaAI, a Mila based startup.
- The goal is to develop machine learning methods for Infant Cry Analysis. (Representation Learning / Interpretability, Efficient Learning, ...)
  Self-Supervised Learning for Cry Analysis,
  ICASSP 2023 workshop paper in the Self-Supervision in Audio, Speech and Beyond Workshop.

SELF-SUPERVISED LEARNING FOR INFANT CRY ANALYSIS

Arsenii Gorin[⋆], Cem Subakan[♮♯♭], Sajjad Abdoli[⋆], Junhao Wang[⋆], Samantha Latremouille[⋆], Charles Onu[⋆♭]

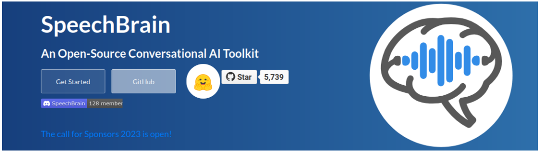


The paper `https://arxiv.org/abs/2305.01578`

Baby Identification Challenge: CryCeleb

The CryCeleb2023 Challenge!

SpeechBrain



Recap

- ASR: We have covered different modern ASR techniques. / On a couvert different techniques de reconnaissance vocale moderne.
  - Encoder-decoder with RNN, transformer, CTC, combination.
- TTS: Conceptually similar sequence-to-sequence techniques as ASR. / Des techniques qui ressemble en concepte à ceux de ASR.
- Source separation/enhancement / Séparation de sources, amélioration du son
  - End-to-end separation, going towards real-life mixtures
- Learning text-audio representations, zero-shot audio classification
- Interpretability for Audio

Suggested Reading

- ASR
  - Book on Speech Processing: https://web.stanford.edu/~jurafsky/slp3/
  - Listen-Attend-Spell: https://arxiv.org/pdf/1508.01211.pdf
  - On CTC: https://www.cs.toronto.edu/~graves/icml_2006.pdf, https://distill.pub/2017/ctc/
- TTS
  - Tacotron: https://arxiv.org/pdf/1703.10135.pdf
  - Transformer TTS: https://arxiv.org/pdf/1809.08895.pdf
  - Tacotron2: https://arxiv.org/pdf/1712.05884.pdf
  - ECAPA-TDNN: https://arxiv.org/pdf/2005.07143.pdf
- Speech Separation
  - Sepformer: https://arxiv.org/abs/2010.13154, https://arxiv.org/abs/2202.02884
  - REAL-M: https://arxiv.org/pdf/2110.10812.pdf
- Cross Model Representations
  - CLAP: https://arxiv.org/pdf/2206.04769.pdf
  - UI-CLAP: https://arxiv.org/abs/2305.01864
- Interpretability for Audio
  - L2I: https://arxiv.org/pdf/2202.11479v2.pdf
  - PIQ: ttps://arxiv.org/pdf/2303.12659.pdf

Next week

- It's your turn now. Don't forget to sign-up:
  `https://docs.google.com/spreadsheets/d/1uZYn_RLkZ_CpQxXTRgoRwZ6b8PdrUtWsZTI1aD-L7Hs/edit?usp=sharing`
    - Le show est à vous maintenant!

Thanks!/Merci!