Automatic Speech Recognition (ASR)

Tal Rosenwein

Outline

- E2E framework
- Timeline
- Benchmarks
- Current Trends

Outline

- **E2E framework**
- Timeline
- Benchmarks
- Current Trends

ASR Models - E2E - Why?

Traditional methods:

- Require a significant amount of **domain specific knowledge**
- acoustic model, pronunciation model, language model are **trained separately** using **different objectives**
- **Complex systems** that contains lots of sub-modules
- Rely on **highly accurate human annotations** that are **costly** (and tend to human errors)

Source 1, 2

ASR Models - E2E - What?

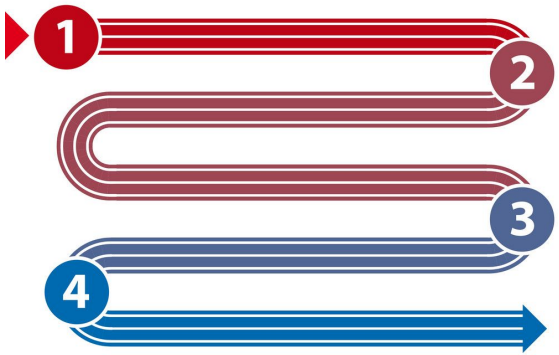
Our **goal** is to achieve **SOTA performances** using **minimal annotation effort** while maintaining the concept of **simplicity**

Source 1

ASR Models - E2E - How?

Jointly train the acoustic, pronunciation and language models with a single objective that **does not enforce a predefined audio-text alignment.**

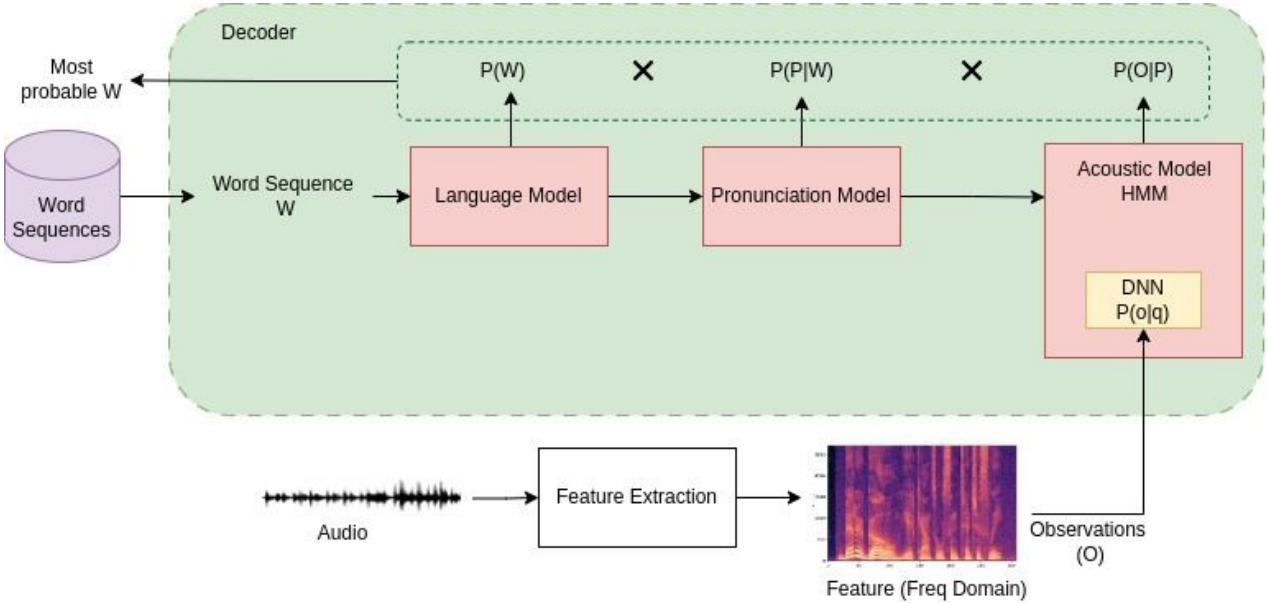
Traditional Methods



End-2-End Methods

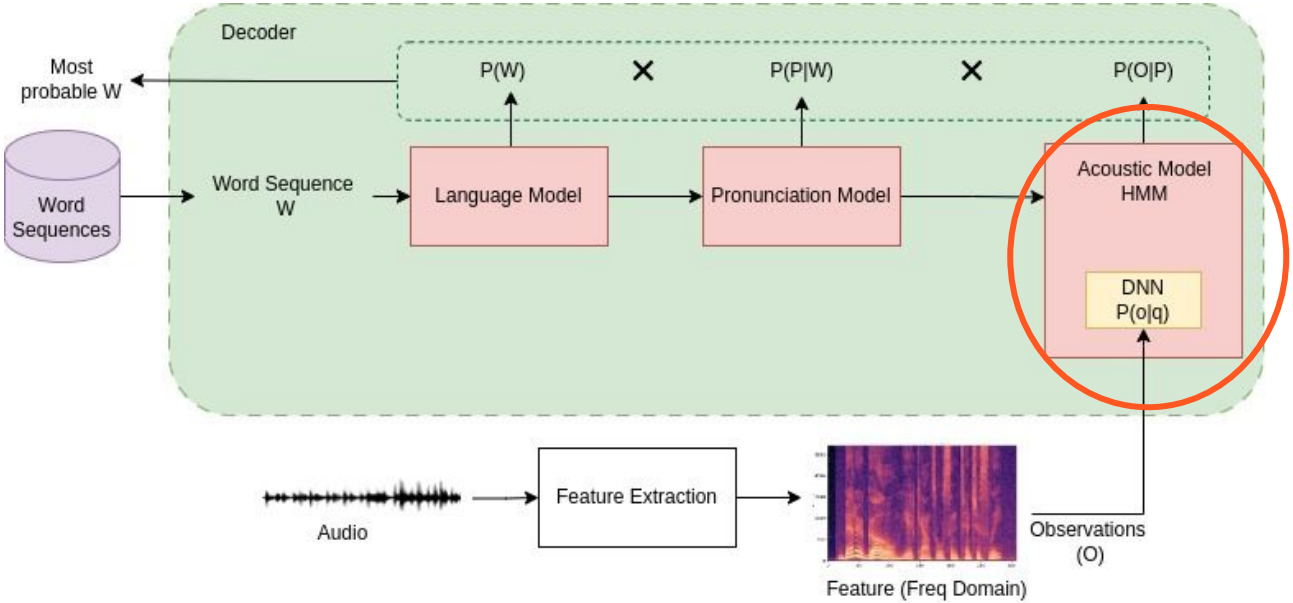
Source 1, 2

ASR Models - E2E - How?



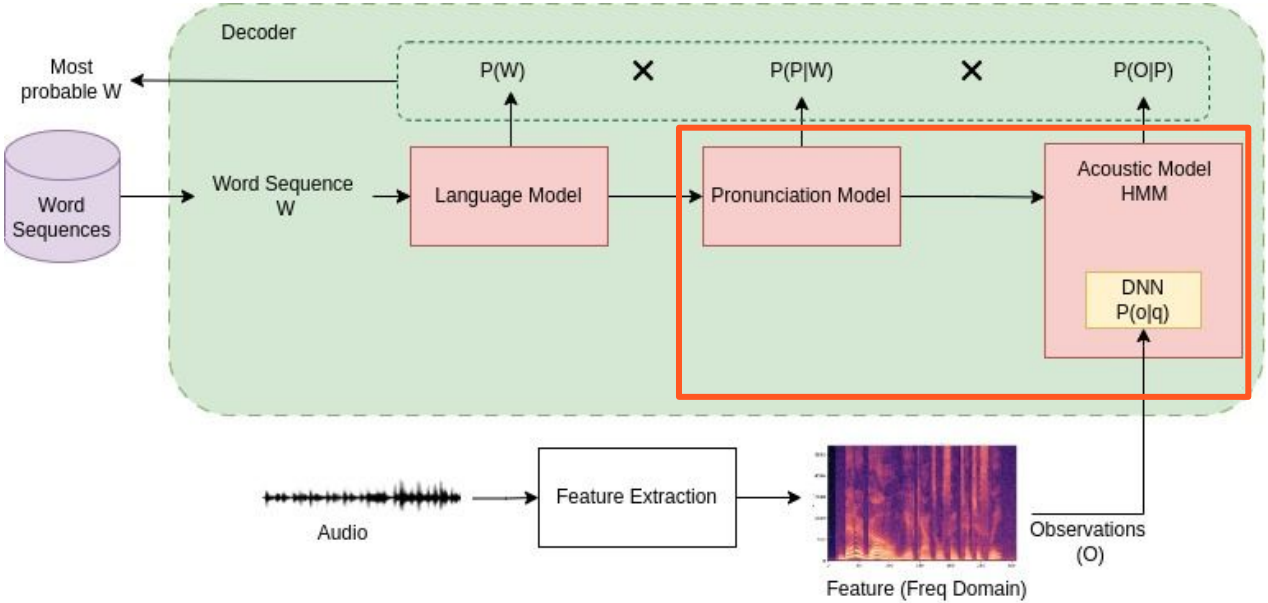
Source 1, 2

ASR Models - E2E - How?



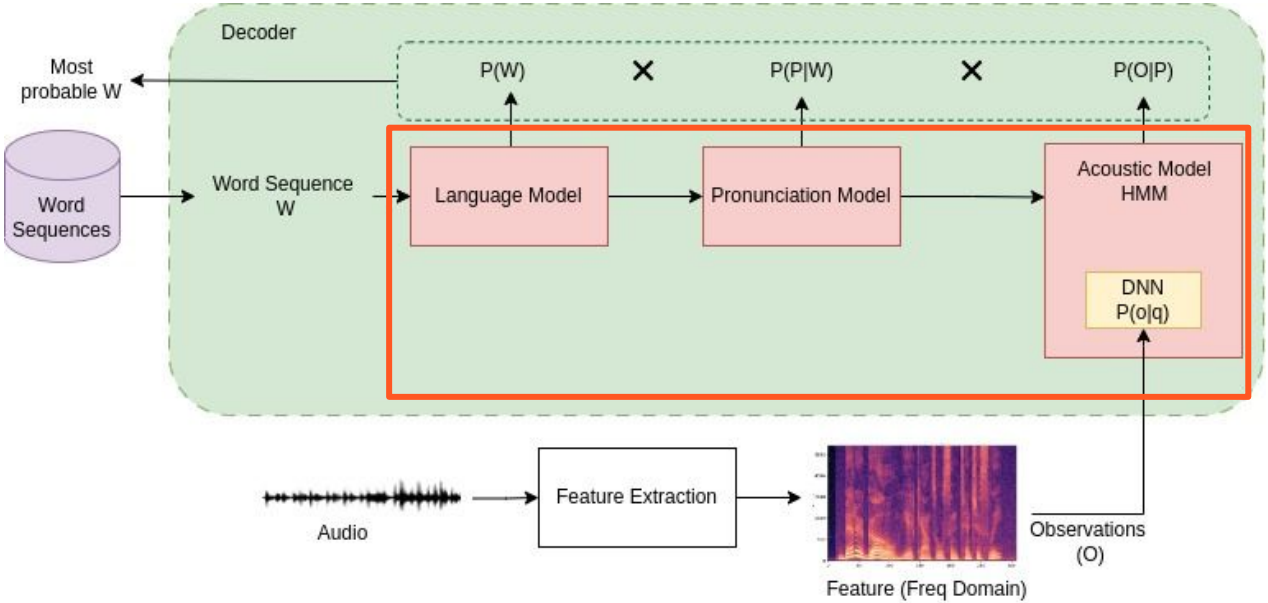
Source 1, 2

ASR Models - E2E - How?



Source 1, 2

ASR Models - E2E - How?



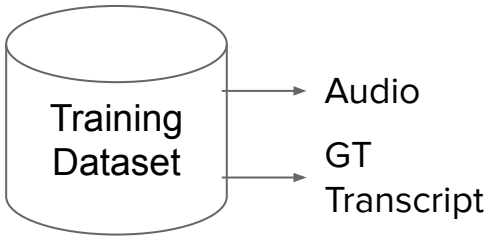
Source 1, 2

ASR Models - E2E - Training Pipeline

Train / Inference

ASR Models - E2E - Training Pipeline

To train the acoustic model, we need to construct a training dataset of audio-transcript pairs. Each pair consists of an audio segment and its corresponding ground truth label (transcript).

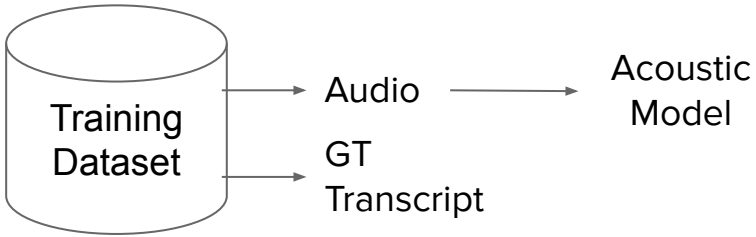


ASR Models - E2E - Training Pipeline

To train the acoustic model, we need to construct a training dataset of audio-transcript pairs. Each pair consists of an audio segment and its corresponding ground truth label (transcript).

Acoustic model training is done using an iterative optimization learning process:

- The audio segment feeds the acoustic AI model as input.

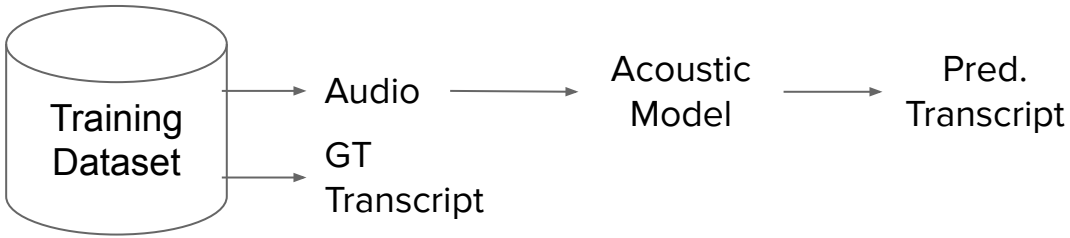


ASR Models - E2E - Training Pipeline

To train the acoustic model, we need to construct a training dataset of audio-transcript pairs. Each pair consists of an audio segment and its corresponding ground truth label (transcript).

Acoustic model training is done using an iterative optimization learning process:

- The audio segment feeds the acoustic AI model as input.
- The model predicts a transcript for that input.

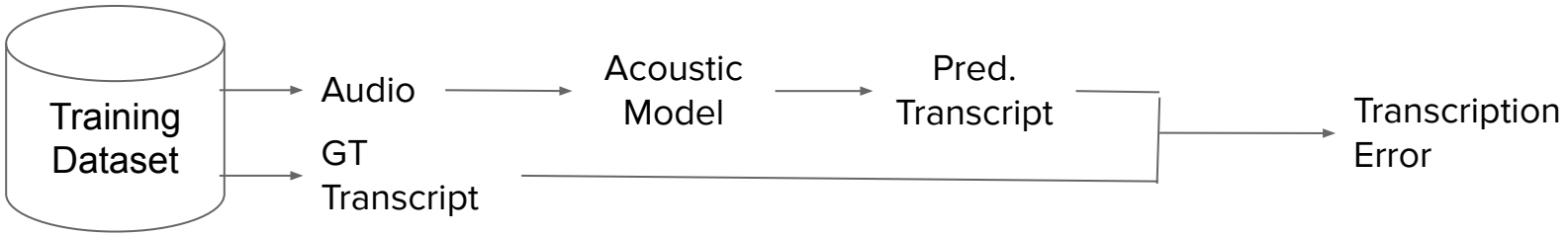


ASR Models - E2E - Training Pipeline

To train the acoustic model, we need to construct a training dataset of audio-transcript pairs. Each pair consists of an audio segment and its corresponding ground truth label (transcript).

Acoustic model training is done using an iterative optimization learning process:

- The audio segment feeds the acoustic AI model as input.
- The model predicts a transcript for that input.
- Transcription error is calculated.

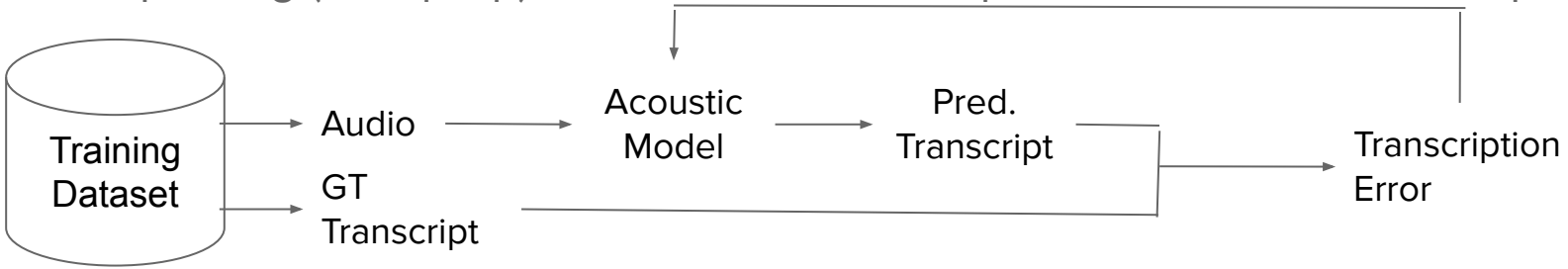


ASR Models - E2E - Training Pipeline

To train the acoustic model, we need to construct a training dataset of audio-transcript pairs. Each pair consists of an audio segment and its corresponding ground truth label (transcript).

Acoustic model training is done using an iterative optimization learning process:

- The audio segment feeds the acoustic AI model as input.
- The model predicts a transcript for that input.
- Transcription error is calculated.
- Updating (backprop) the acoustic model's params to reduce transcription error.



ASR Models - E2E - Training Pipeline

To train the acoustic model, we need to construct a training dataset of audio-transcript pairs. Each pair consists of an audio segment and its corresponding ground truth label (transcript).

Acoustic model training is done using an iterative optimization learning process:

- The audio segment feeds the acoustic AI model as input.
- The model predicts a transcript for that input.
- Transcription error is calculated.
- Updating (backprop) the acoustic model's params to reduce transcription error.

This process has millions of iterations in which the predicted transcripts are improved over time.

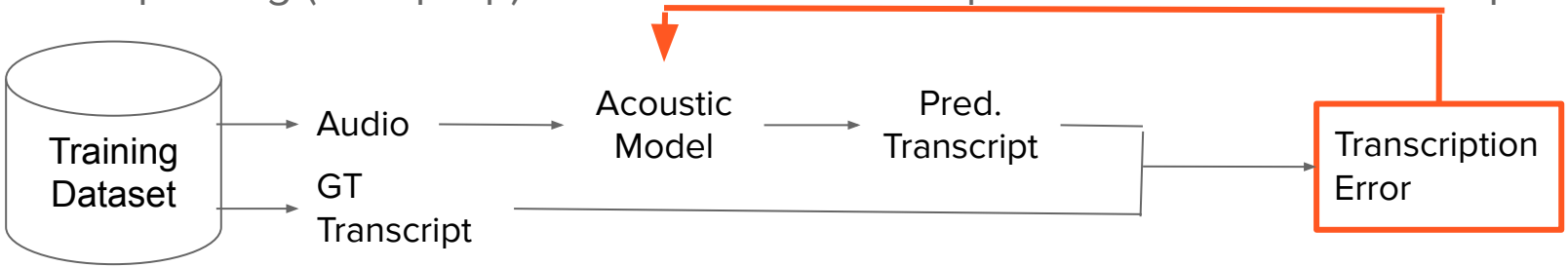
ASR Models - E2E - Training Pipeline

To train the acoustic model, we need to construct a training dataset of audio-transcript pairs. Each pair consists of an audio segment and its corresponding ground truth label (transcript).

Acous

- T
- T
- Transcription error is calculated.
- Updating (backprop) the acoustic model's params to reduce transcription error.

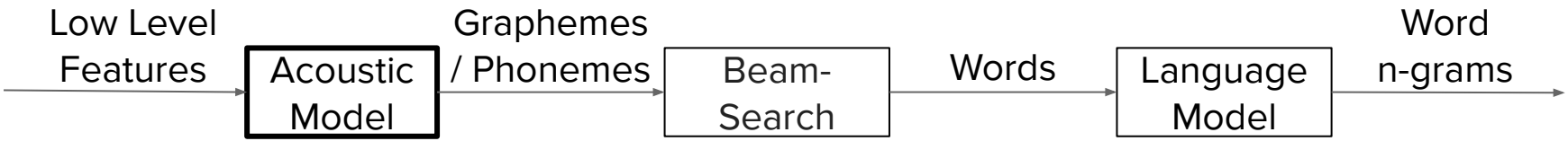
We will discuss mainly on the loss function.
Not the acoustic model's architecture

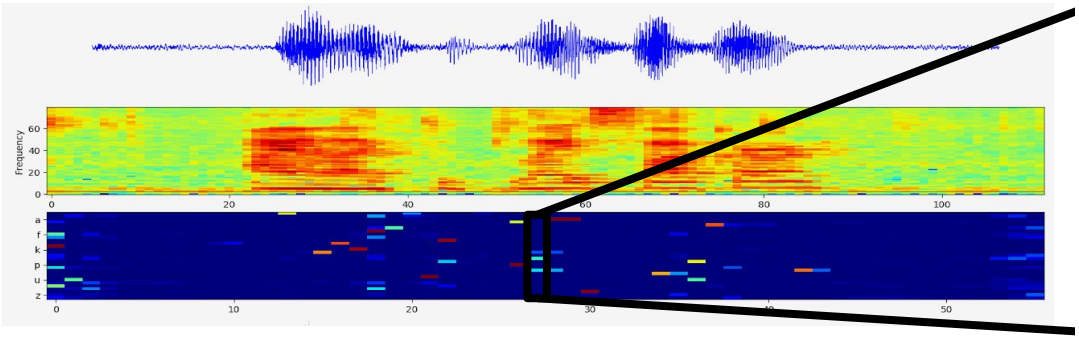


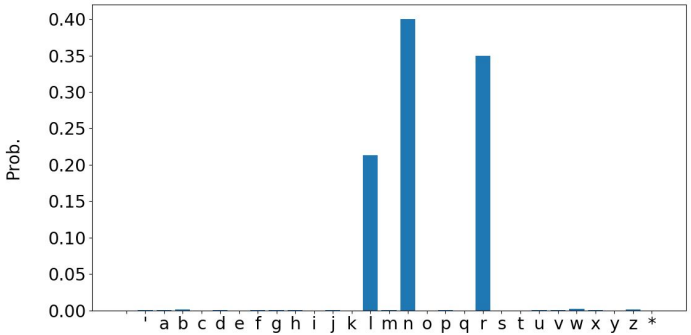
ASR Models - E2E - Inference

Train / Inference

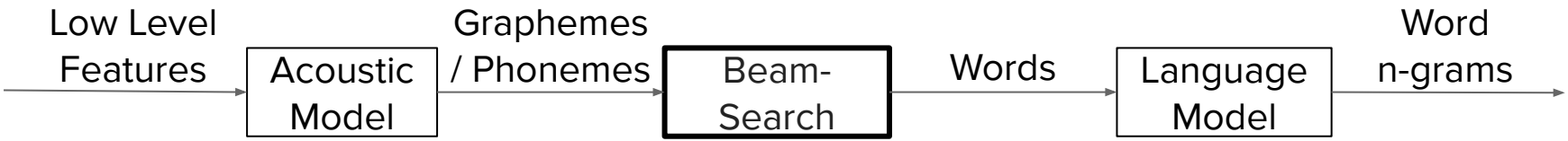
ASR Models - E2E - Inference





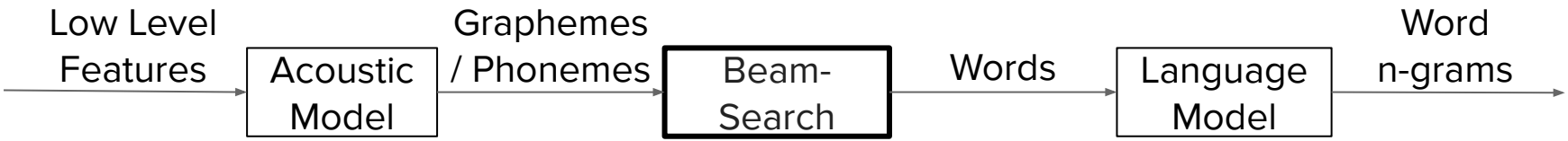


ASR Models - E2E - Inference

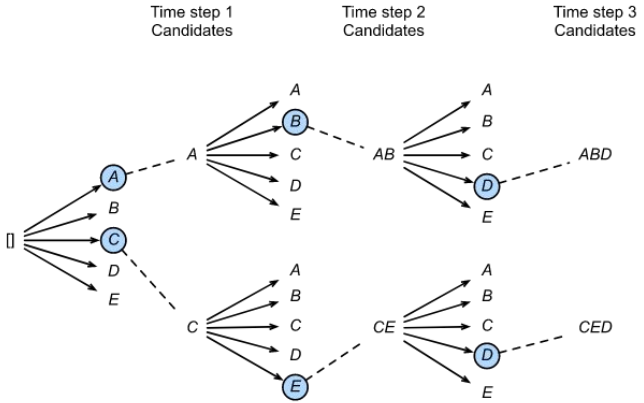


Decoding the acoustic model's output:
Argmax decoder Vs Beam search decoder

ASR Models - E2E - Inference

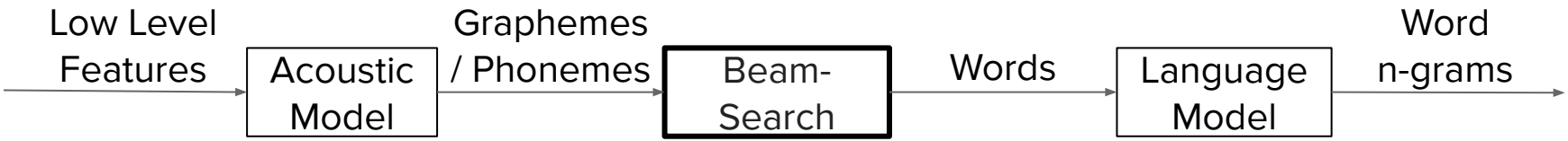


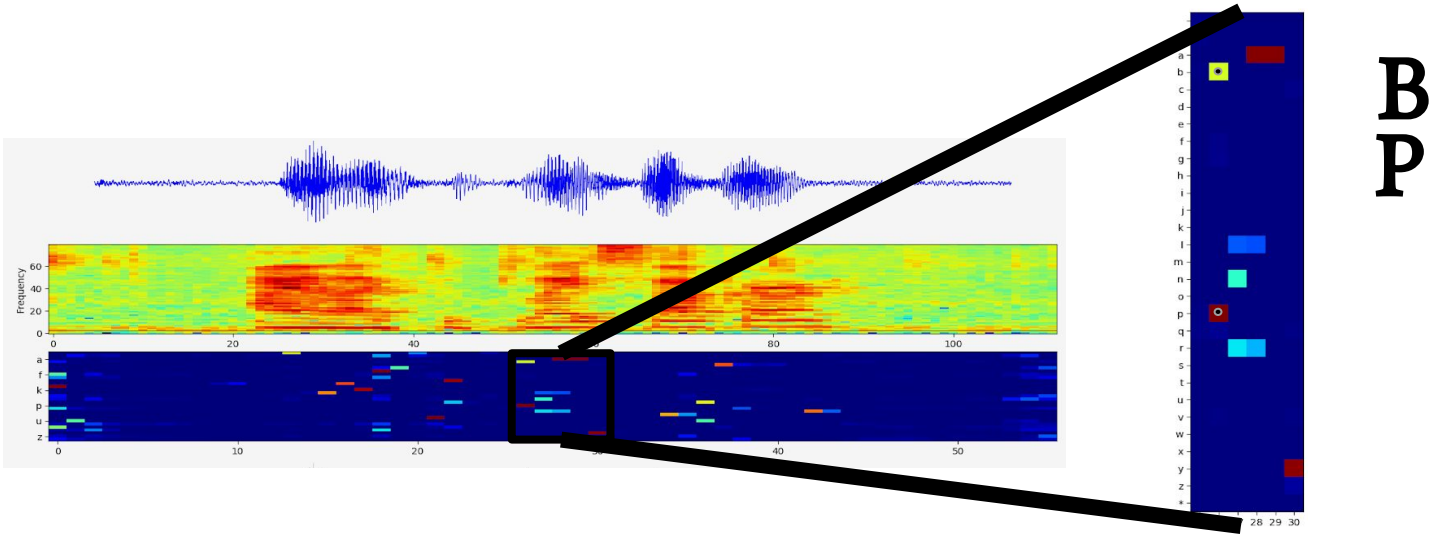
Beam-Search is a (breadth-first) heuristic **search** algorithm that explores a graph by **expanding the most promising nodes in a limited set**



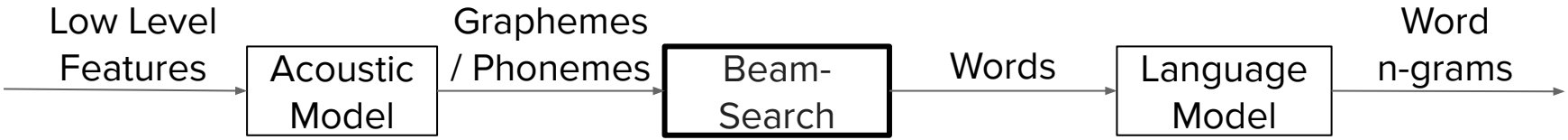
Source 1

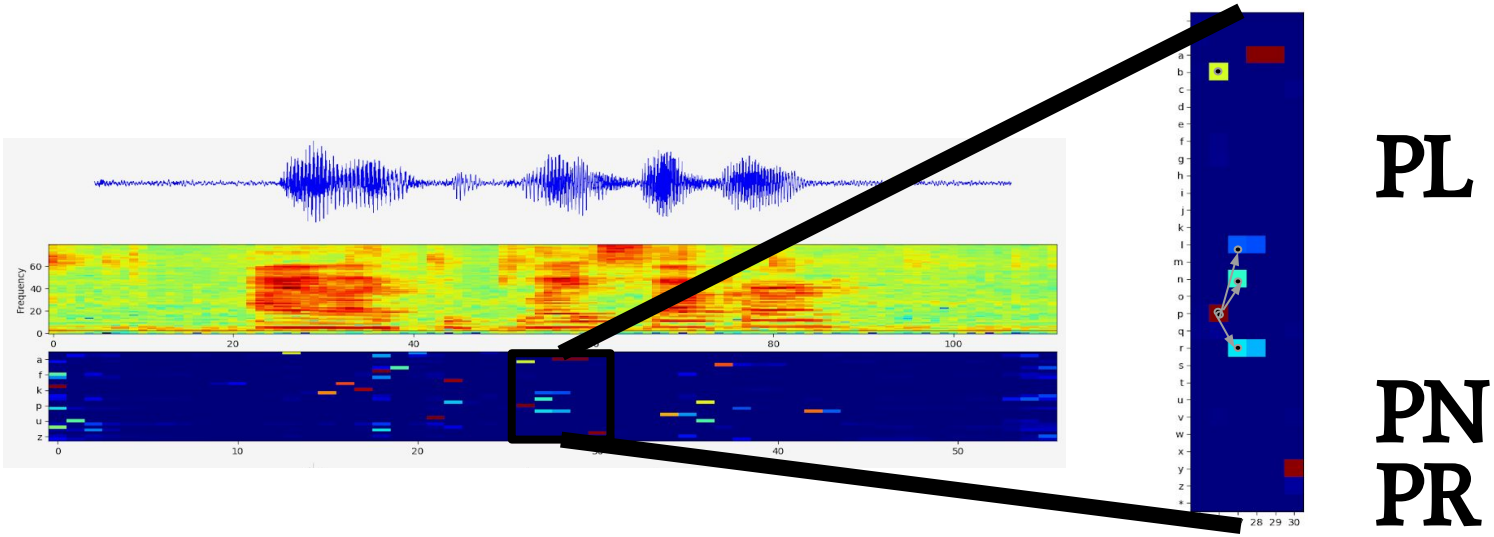
ASR Models - E2E - Inference



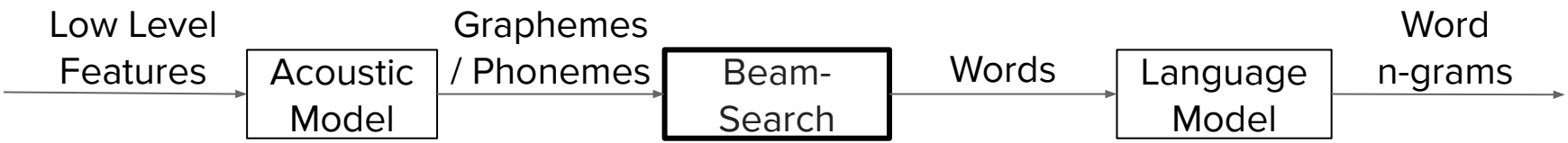


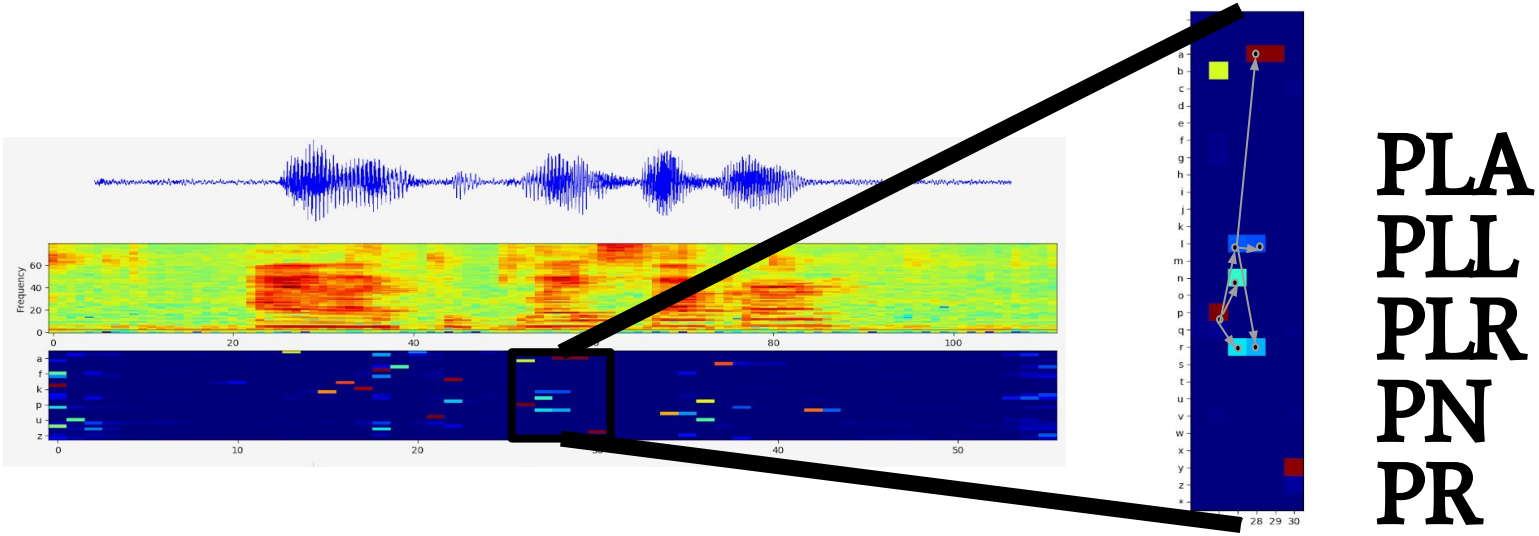
ASR Models - E2E - Inference



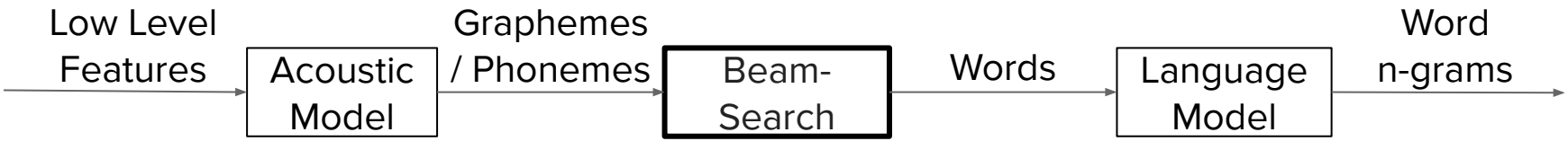


ASR Models - E2E - Inference



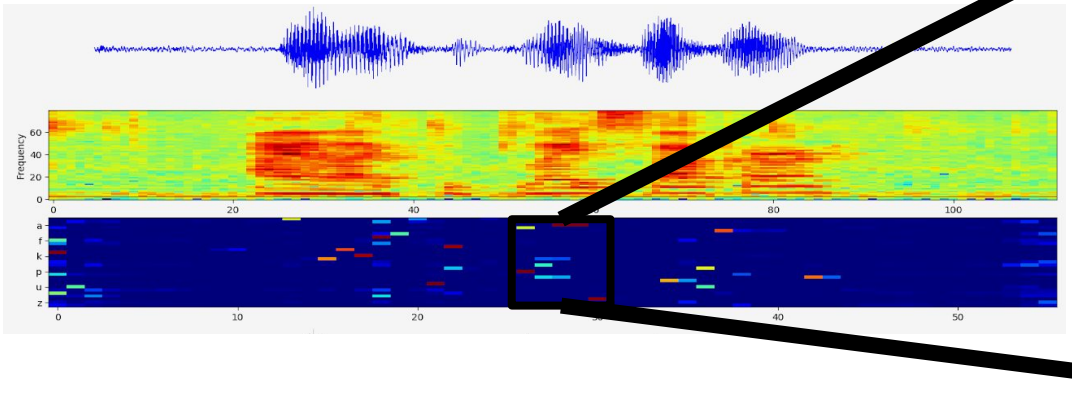


ASR Models - E2E - Inference



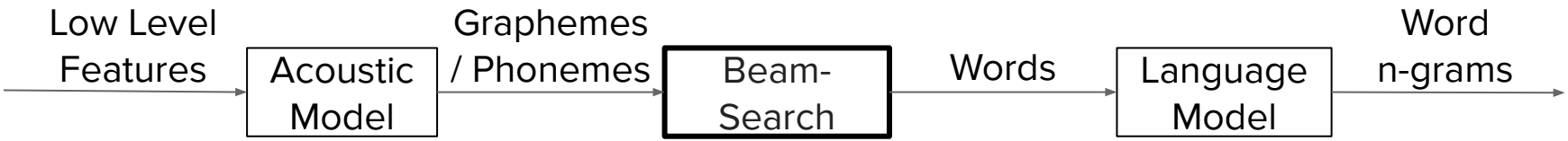
F*(aabbaa) = *F(abaaa) = aba

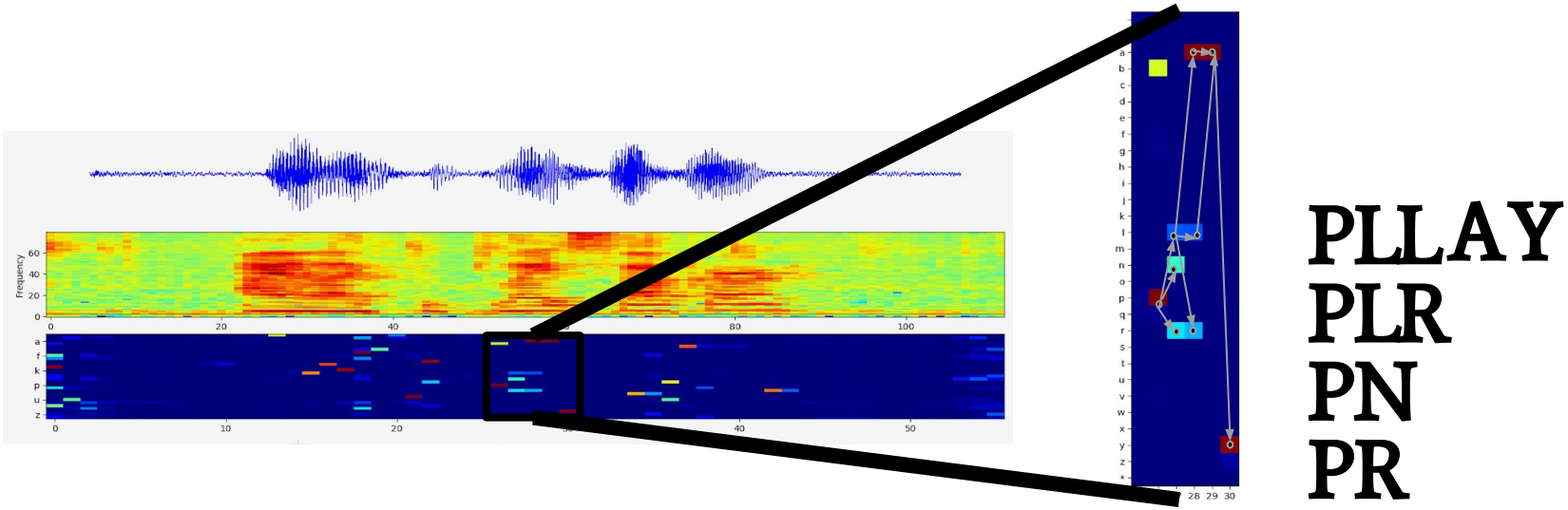
P(“pla”) = P(“plaa”) + P(“plla”)



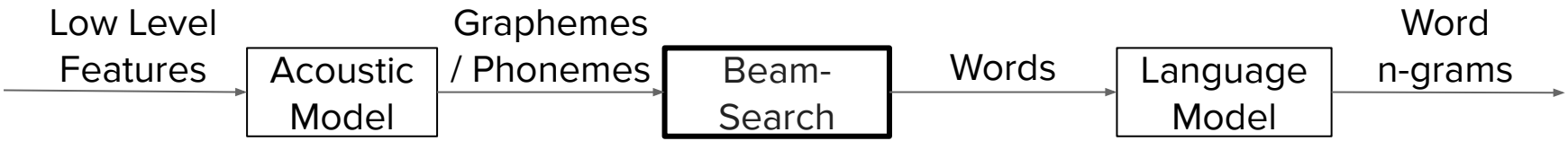
PLA A
PLLA
PLR
PN
PR

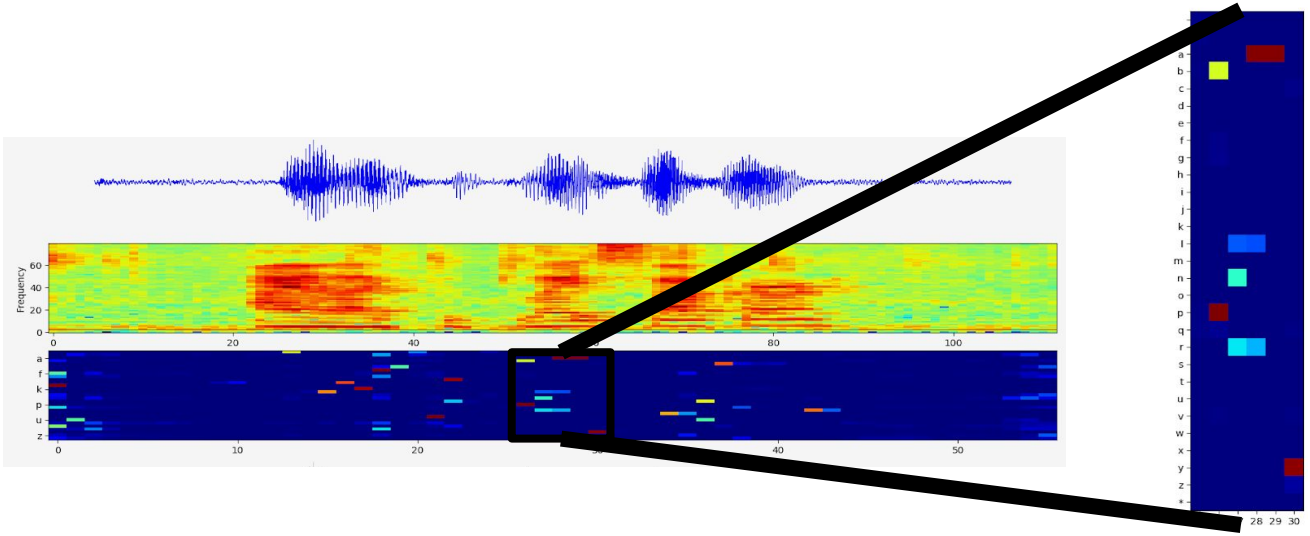
ASR Models - E2E - Inference





ASR Models - E2E - Inference





Beam

play

pray

pnay

----

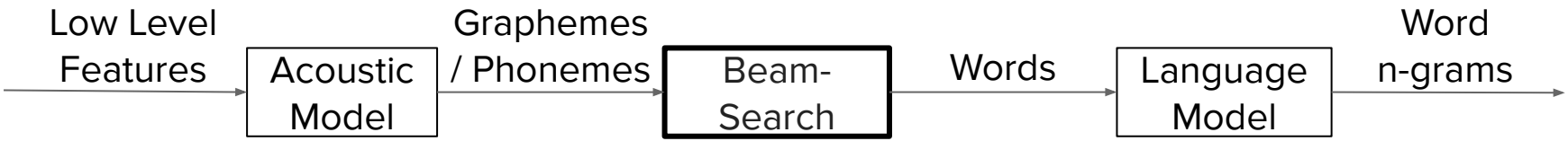
bray

plray

bnay

blay

ASR Models - E2E - Inference

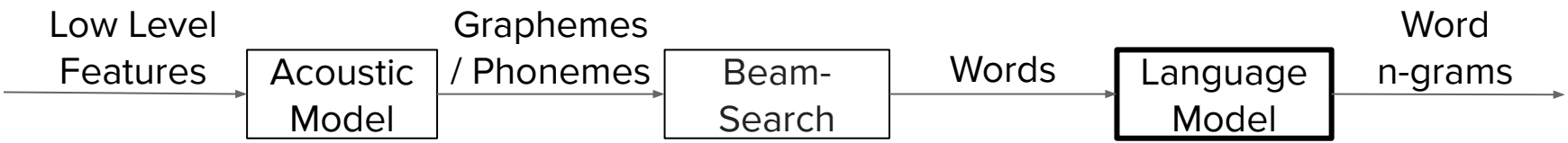


P("play")

P("pray")

P("pnay") - Not In Dictionary

ASR Models - E2E - Inference



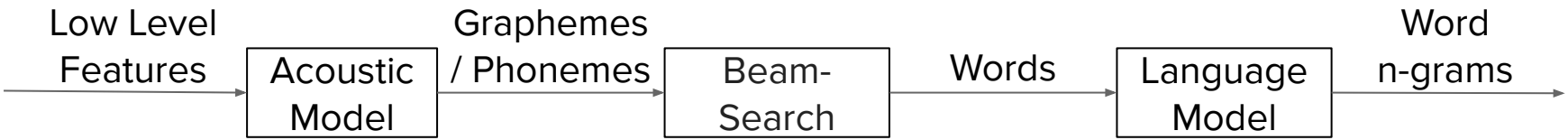
Language model rescores the word sequences and returns the most probable text

I like to play soccer > I like to pray soccer

P("I like to play soccer") = P("I like to") * P("like to play") * P("to play soccer") = 0.9 * 0.8 * 0.85 = **0.612**

P("I like to pray soccer") = P("I like to") * P("like to pray") * P("to pray soccer") = 0.9 * 0.85 * 0.01 = **0.008**

ASR Models - E2E - Inference



Acoustic-Model Captures Pronunciation But Not Spelling / Context

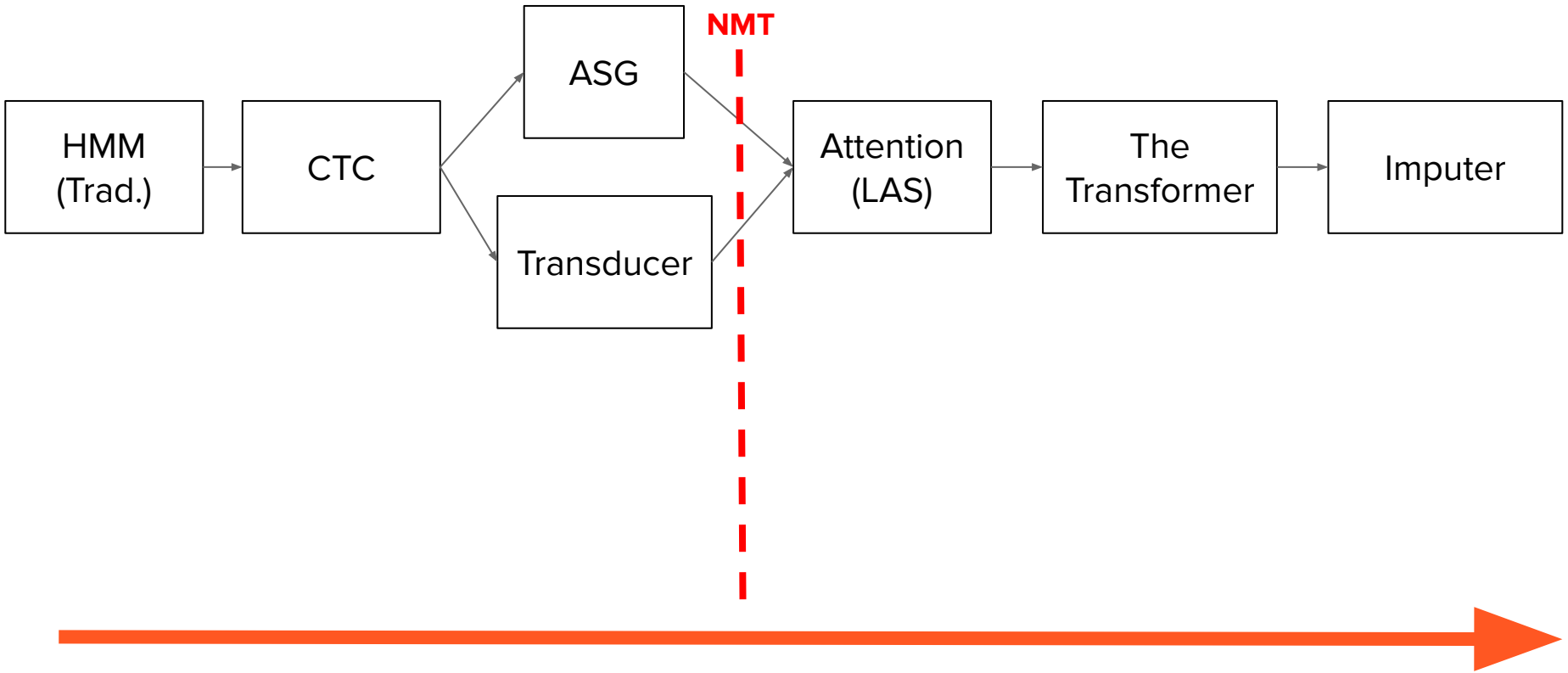
Beam-Search Captures Spelling But Not Context

Language-Model Captures Context

Outline

- E2E framework
- **Timeline**
  - GMM-HMM & DNN-HMM
  - CTC
  - ASG
  - LAS
  - RNN-T
  - Transformer (Conformer)
  - Imputer
- Benchmarks
- Current Trends

ASR Models - Timeline



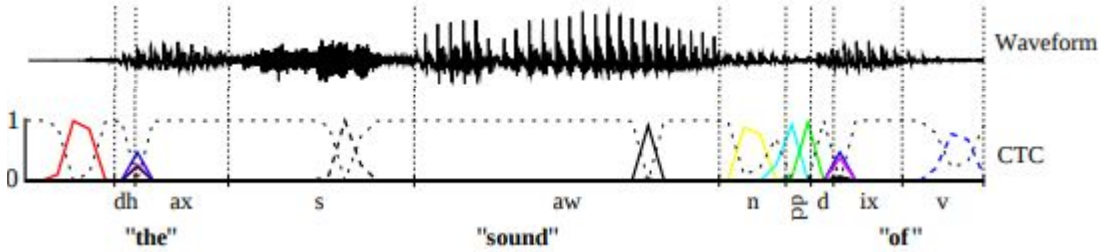
Outline

- E2E framework
- **Timeline**
  - GMM-HMM & DNN-HMM
  - CTC
  - ASG
  - LAS
  - RNN-T
  - Transformer (Conformer)
  - Imputer
- Benchmarks
- Current Trends

ASR Models - Timeline

ASR Models - CTC

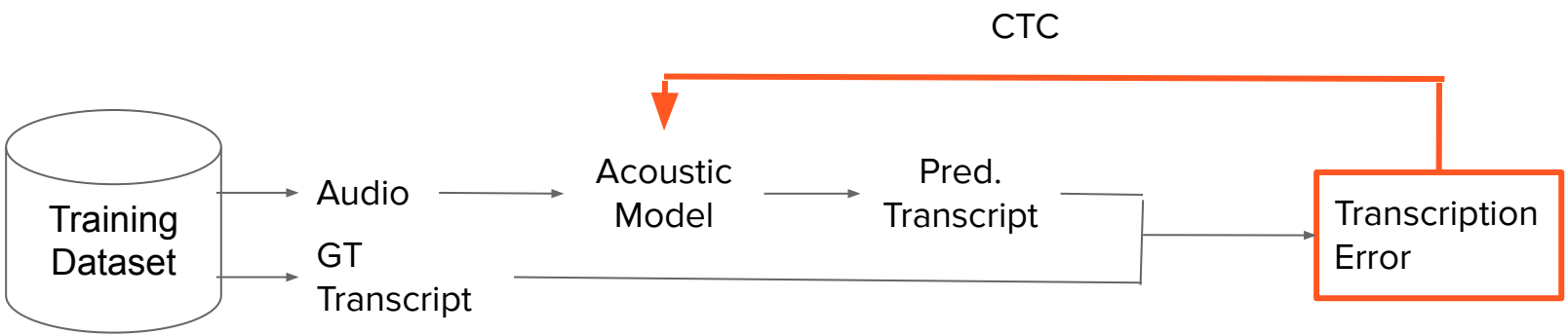
Connectionist Temporal Classification (CTC) enables end-to-end training of acoustic models on unsegmented sequences



Connectionist Temporal Classification: Labelling Unsegmented Sequence Data with Recurrent Neural Networks, A. Graves & J. Schmidhuber, ICML 2006

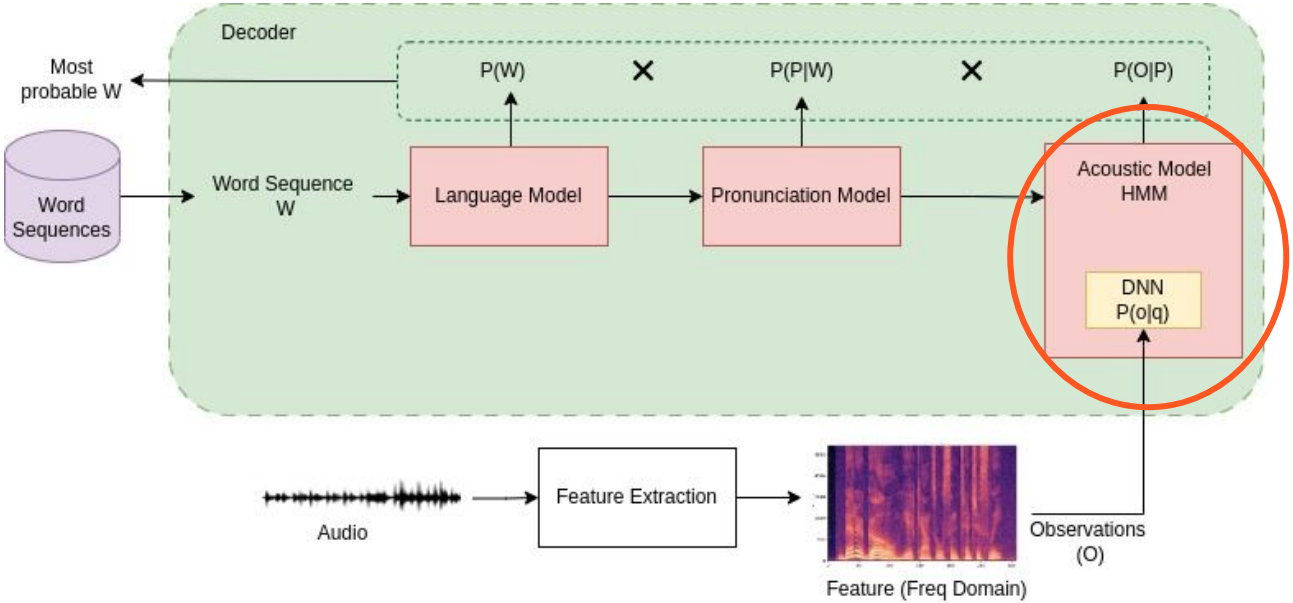
ASR Models - CTC

Connectionist Temporal Classification (CTC) enables end-to-end training of acoustic models on unsegmented sequences



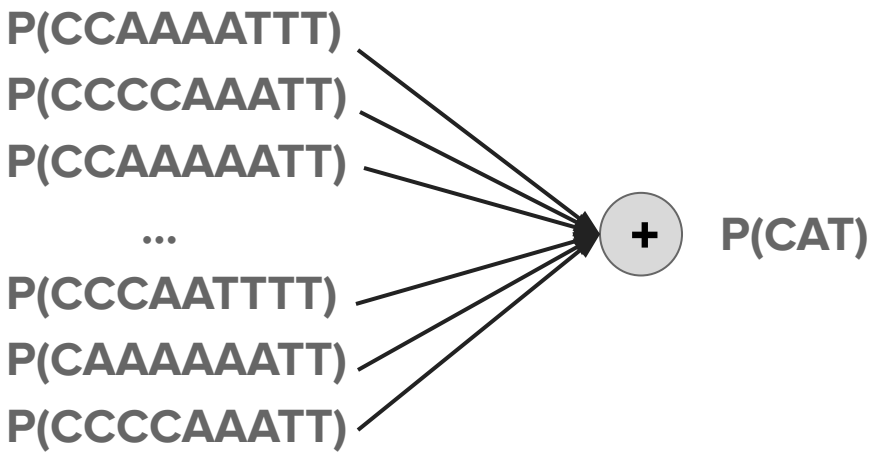
Connectionist Temporal Classification: Labelling Unsegmented Sequence Data with Recurrent Neural Networks, A. Graves & J. Schmidhuber, ICML 2006

ASR Models - CTC



Source 1, 2

ASR Models - CTC



ASR Models - CTC: Train

Annotations:

- L: Alphabet:
- $L' = L \bigcup \{blank\}$
- y_k^t: Probability of observing label k at time t
- $\pi = L^{'T}$: Path - a sequence of length T. E.g: $-aa-abb$

$$p(\pi|x) = \prod_{t=1}^{T} y_{\pi_t}^t, \forall \pi \epsilon L^{'T}$$

Conditional independence assumption between model's outputs

ASR Models - CTC: Train

Define Many-To-One Map: remove all blanks and repeated labels

$$B : L'^{T} \mapsto L^{\leqslant T}$$

$$B(a--ab-) = B(-aa-abb) = aab$$

'Blank' state models:

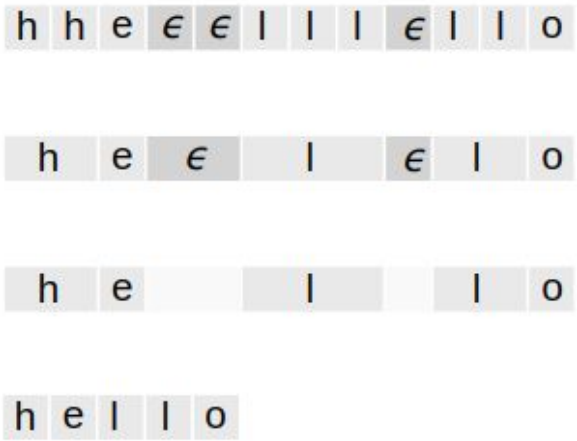
- "garbage" frames
- separation between two identical consecutive letters

CTC Maximizes:

$$p(\mathbf{l}|x) = \sum_{\pi \epsilon B^{-1}(\mathbf{l})} p(\pi|x), \mathbf{l} \epsilon L^{\leq T}$$

ASR Models - CTC: Train

Define Many-To-One Map: remove all blanks and repeated labels



'Blank' state mo

- "garbag
- separati

CTC Maximizes:

$$p(\mathbf{l}|x) = \sum_{\pi \epsilon B^{-1}(\mathbf{l})} p(\pi|x), \mathbf{l} \epsilon L^{\leq T}$$

Source 1

ASR Models - CTC: Train

Define Many-To-One Map: remove all blanks and repeated labels

$$B : L'^{T} \mapsto L^{\leqslant T}$$

$$B(a--ab-) = B(-aa-abb) = aab$$

'Blank' state models:

- "garbage" frames
- separation between two identical consecutive letters

CTC Maximizes:

$$p(\mathbf{l}|x) = \sum_{\pi \epsilon B^{-1}(\mathbf{l})} p(\pi|x), \mathbf{l} \epsilon L^{\leq T}$$

ASR Models - CTC: Train

Define Many-To-One Map: remove all blanks and repeated labels

$$B : L'^{T} \mapsto L^{\leqslant T}$$

$$B(a--ab-) = B(-aa-abb) = aab$$

'Blank' state models:

- "garbage" frames
- separation between two identical consecutive letters

CTC Maximizes:

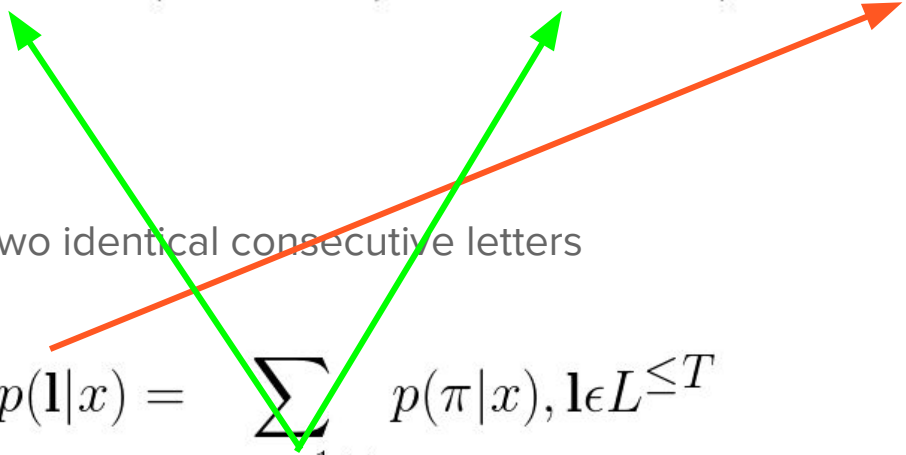
$$p(\mathbf{l}|x) = \sum_{\pi \epsilon B^{-1}(\mathbf{l})} p(\pi|x), \mathbf{l} \epsilon L^{\leq T}$$

ASR Models - CTC: Train

Define Many-To-One Map: remove all blanks and repeated labels

$$B : L'^{T} \mapsto L^{\leqslant T}$$

$$B(a--ab-) = B(-aa-abb) = aab$$



'Blank' state models:

- "garbage" frames
- separation between two identical consecutive letters

CTC Maximizes:

$$p(\mathbf{l}|x) = \sum_{\pi \epsilon B^{-1}(\mathbf{l})} p(\pi|x), \mathbf{l} \epsilon L^{\leq T}$$

ASR Models - CTC: Train

Define Many-To-One Map: remove all blanks and repeated labels

$$B : L'^{T} \mapsto L^{\leqslant T}$$

$$B(a--ab-) = B(-aa-abb) = aab$$

1 2 3 4 5 6

'Blank' state models:

- "garbage" frames
- separation between two identical consecutive letters

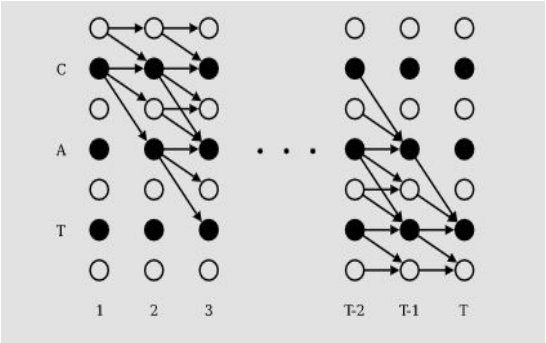
CTC Maximizes:

$$p(\mathbf{l}|x) = \sum_{\pi \epsilon B^{-1}(\mathbf{l})} p(\pi|x), \mathbf{l} \epsilon L^{\leq T}$$

ASR Models - CTC: Train

$$p(\mathbf{l}|x) = \sum_{\pi \epsilon B^{-1}(\mathbf{l})} p(\pi|x), \mathbf{l} \epsilon L^{\leq T}$$

$p(\mathbf{l}|x)$ is **efficiently calculated** using a Forward Backward Algorithm (**dynamic programming**)



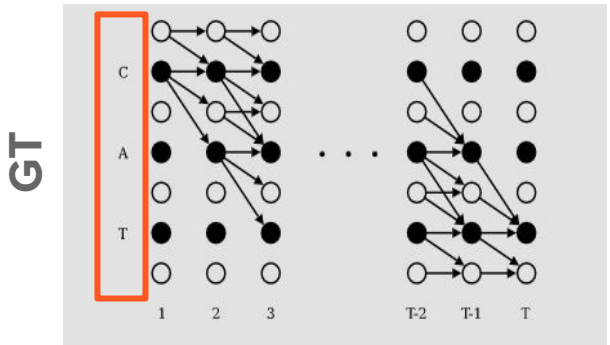
Towards End-to-End Speech Recognition with Recurrent Neural Networks (Graves & Jaitly 2014)

ASR Models - CTC: Train

$$p(\mathbf{l}|x) = \sum_{\pi \epsilon B^{-1}(\mathbf{l})} p(\pi|x), \mathbf{l} \epsilon L^{\leq T}$$

$p(\mathbf{l}|x)$ is **efficiently calculated** using a Forward Backward Algorithm (**dynamic programming**)

Define a new sequence: **CAT → ^C^A^T^**



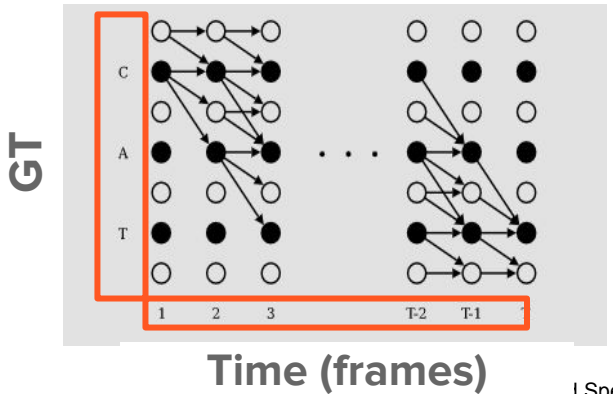
Towards End-to-End Speech Recognition with Recurrent Neural Networks (Graves & Jaitly 2014)

ASR Models - CTC: Train

$$p(\mathbf{l}|x) = \sum_{\pi \epsilon B^{-1}(\mathbf{l})} p(\pi|x), \mathbf{l} \epsilon L^{\leq T}$$

$p(\mathbf{l}|x)$ is **efficiently calculated** using a Forward Backward Algorithm (**dynamic programming**)

Define a new sequence: **CAT → ^C^A^T^**



Speech Recognition with Recurrent Neural Networks (Graves & Jaitly 2014)

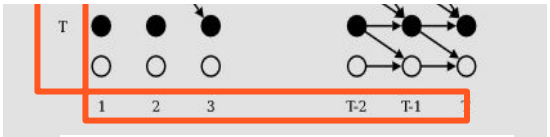
ASR Models - CTC: Train

$$p(\mathbf{l}|x) = \sum_{\pi \epsilon B^{-1}(\mathbf{l})} p(\pi|x), \mathbf{l} \epsilon L^{\leq T}$$

$p(\mathbf{l}|x)$ is **efficiently calculated** using a Forward Backward Algorithm (**dynamic programming**)

Define a new sequence: **CAT → ^C^A^T^**

$\alpha_s(t)$ represents the probability of observing the prefix $S_{1:s}$ by time t



Time (frames)

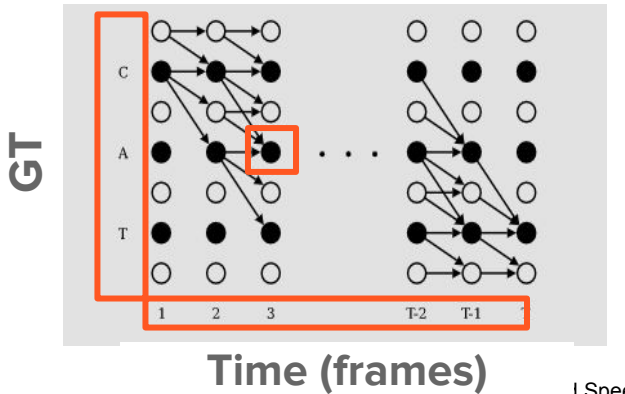
Speech Recognition with Recurrent Neural Networks (Graves & Jaitly 2014)

ASR Models - CTC: Train

$$p(\mathbf{l}|x) = \sum_{\pi \epsilon B^{-1}(\mathbf{l})} p(\pi|x), \mathbf{l} \epsilon L^{\leq T}$$

$\alpha_s(t)$ represents the probability of observing the prefix $S_{1:s}$ by time t

Define a new sequence: **CAT → ^C^A^T^**



$$\alpha_s(t)$$

Speech Recognition with Recurrent Neural Networks (Graves & Jaitly 2014)

ASR Models - CTC: Train

$$p(\mathbf{l}|x) = \sum_{\pi \epsilon B^{-1}(\mathbf{l})} p(\pi|x), \mathbf{l} \epsilon L^{\leq T}$$

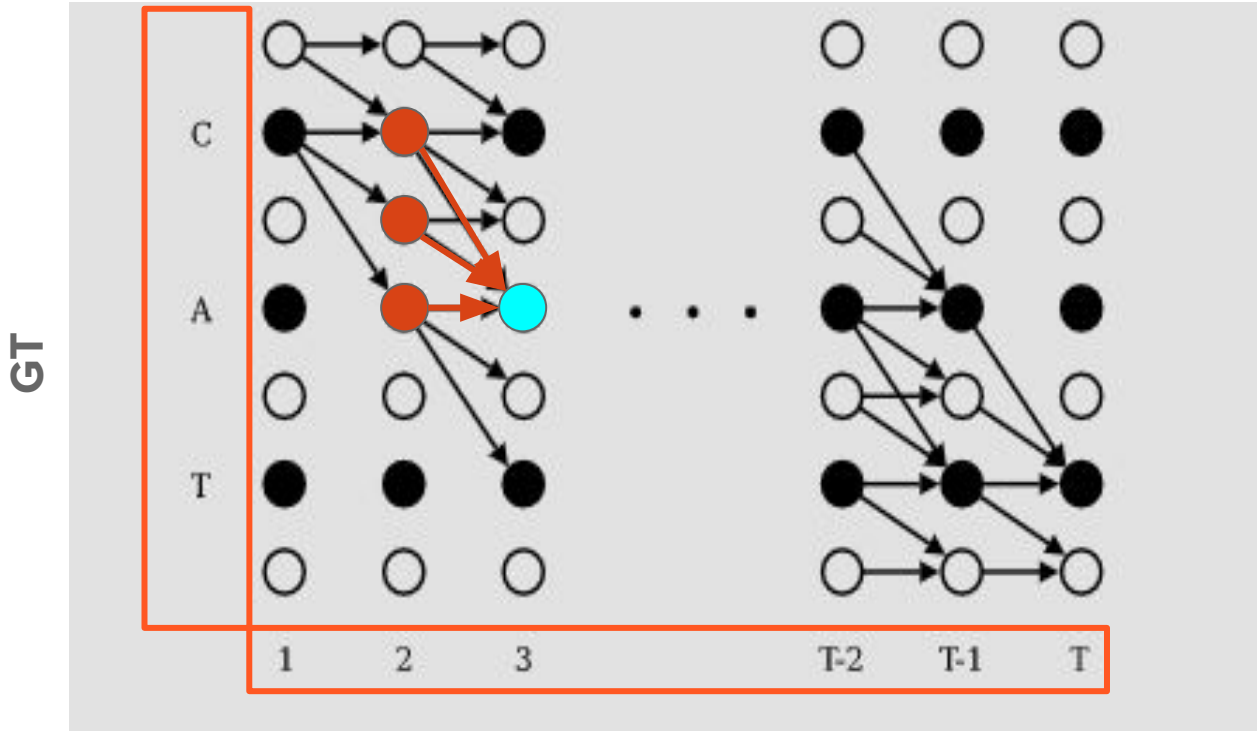
$p(\mathbf{l}|x)$ is **efficiently calculated** using a Forward Backward Algorithm (**dynamic programming**)

Non-Blank symbol

- **Case 1: non identical letters in the GT string- 3 options:**
  - **New letter after prev letter (C → A)**
  - **New letter after blank (C→ ^ → A)**
  - **Same letter twice, collapse will remove it. (A → A)**
- Case 2: GT has 2 repeated letters (e.g., ll as in the word Hello)- 2 options:
  - New letter after blank (L→ ^ → L)
  - Same letter twice, collapse will remove it. (L → L)

ASR Models - CTC: Train

$$\alpha_s(t) = p_t(S_s|X)\left(\alpha_{s-2}(t-1) + \alpha_{s-1}(t-1) + \alpha_s(t-1)\right)$$



ASR Models - CTC: Train

$$p(\mathbf{l}|x) = \sum_{\pi \epsilon B^{-1}(\mathbf{l})} p(\pi|x), \mathbf{l} \epsilon L^{\leq T}$$

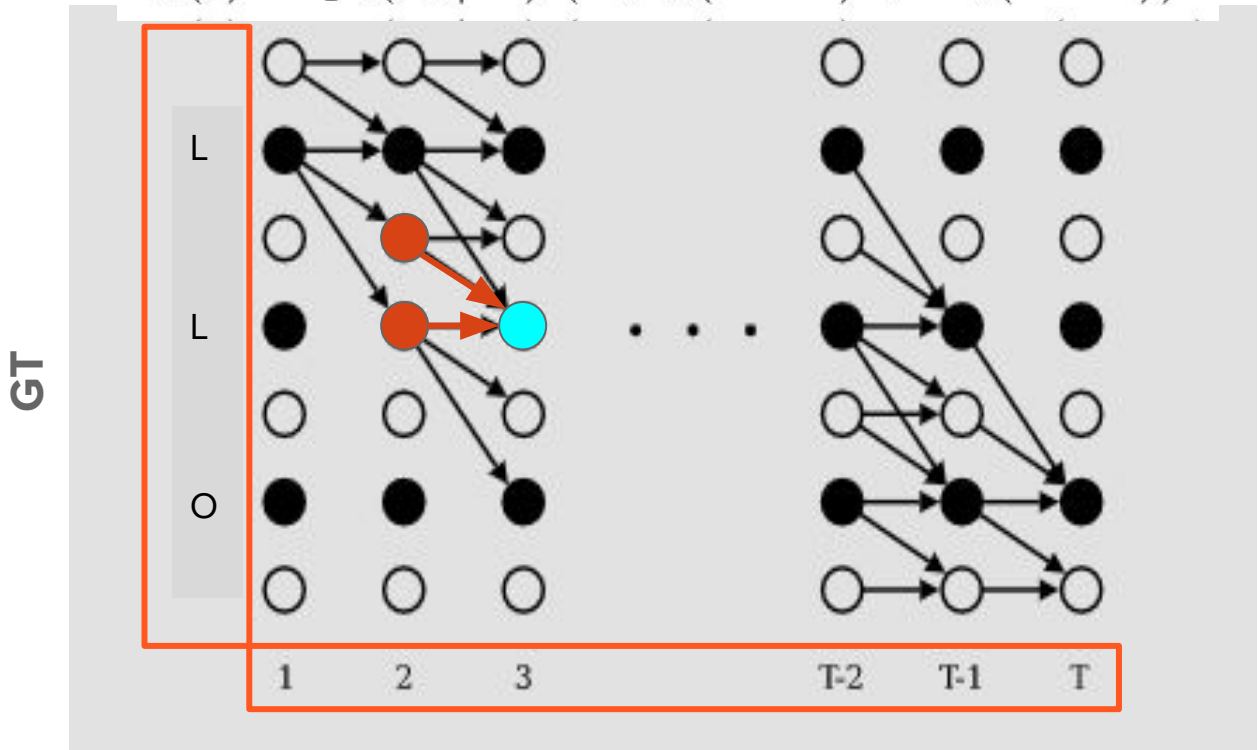
$p(\mathbf{l}|x)$ is **efficiently calculated** using a Forward Backward Algorithm (**dynamic programming**)

Non-Blank symbol

- Case 1: non identical letters in the GT string- 3 options:
  - New letter after prev letter (C → A)
  - New letter after blank (C→ ^ → A)
  - Same letter twice, collapse will remove it. (A → A)
- **Case 2: GT has 2 repeated letters (e.g., ll as in the word Hello)- 2 options:**
  - **New letter after blank (L→ ^ → L)**
  - **Same letter twice, collapse will remove it. (L → L)**

ASR Models - CTC: Train

$$\alpha_s(t) = p_t(S_s|X)\,(\alpha_{s-1}(t-1) + \alpha_s(t-1))$$



ASR Models - CTC: Train

$$p(\mathbf{l}|x) = \sum_{\pi \epsilon B^{-1}(\mathbf{l})} p(\pi|x), \mathbf{l} \epsilon L^{\leq T}$$

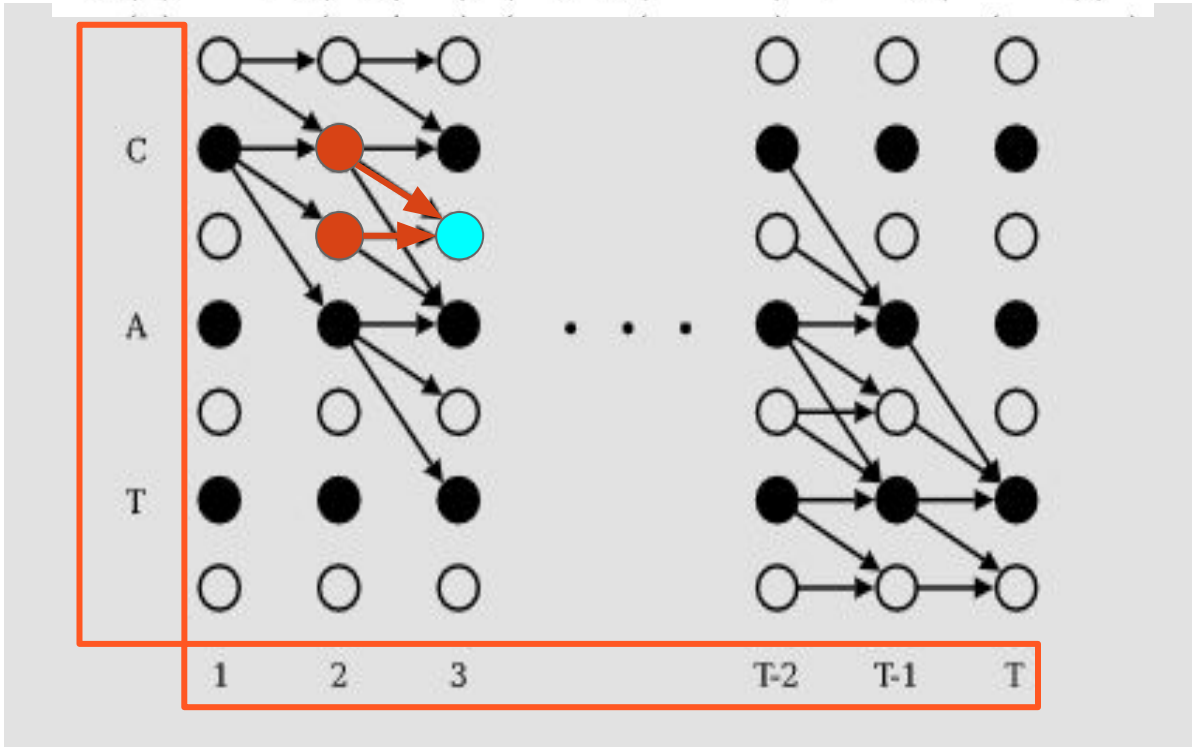
$p(\mathbf{l}|x)$ is **efficiently calculated** using a Forward Backward Algorithm (**dynamic programming**)

Blank symbol

- Single case- 2 options:
  - Blank after non-blank letter (C ➔ ^)
  - Blank after blank (^ ➔ ^)

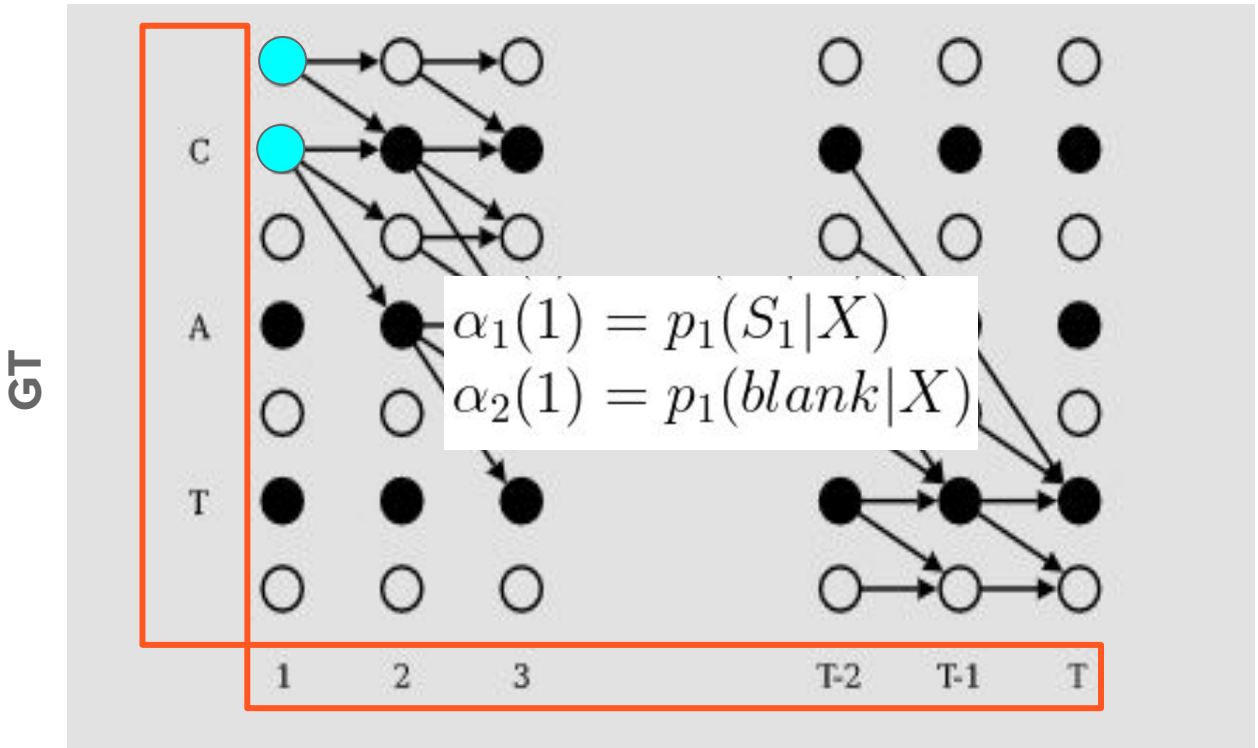
ASR Models - CTC: Train

$$\alpha_s(t) = p_t(S_s|X)\,(\alpha_{s-1}(t-1) + \alpha_s(t-1))$$

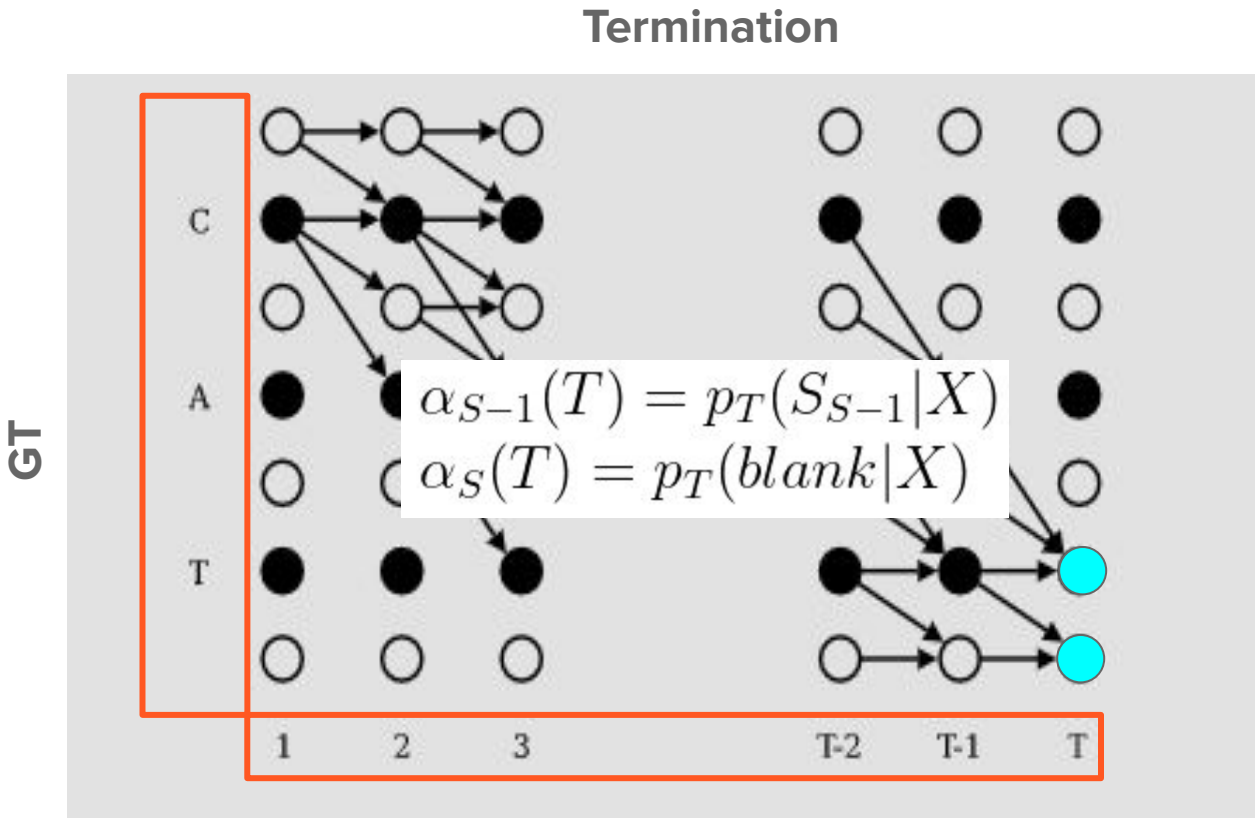


ASR Models - CTC: Train

Initialization



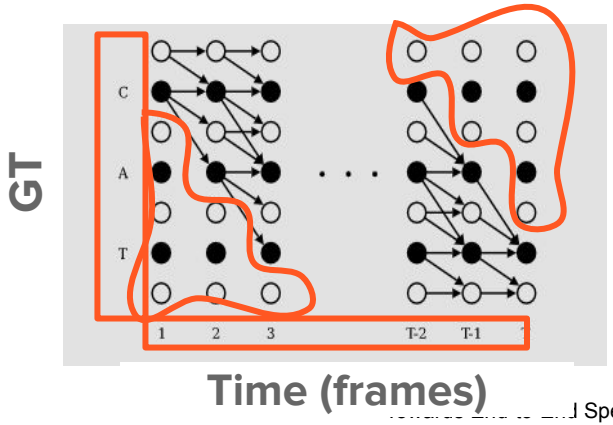
ASR Models - CTC: Train



ASR Models - CTC: Train

$$p(\mathbf{l}|x) = \sum_{\pi \epsilon B^{-1}(\mathbf{l})} p(\pi|x), \mathbf{l} \epsilon L^{\leq T}$$

$p(\mathbf{l}|x)$ is **efficiently calculated** using a Forward Backward Algorithm (**dynamic programming**)



Speech Recognition with Recurrent Neural Networks (Graves & Jaitly 2014)

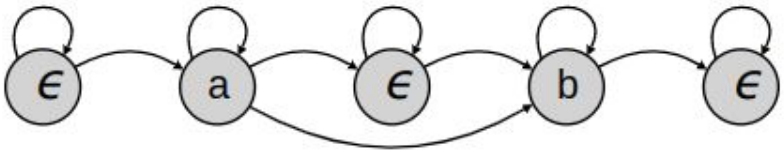
ASR Models - CTC: Train

The **Forward Backward Algorithm** is **fully differentiable.**
hence the acoustic model can be **trained** using **SGD.**

ASR Models - CTC Vs HMM

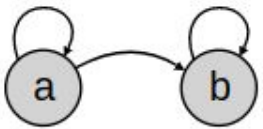
CTC Vs HMM

$$P(O|P) = P(O|Q) = \prod_{t=1}^{T} P(o_t|q_t)$$



CTC HMM: The first two nodes are the starting states and the last two nodes are the final states.

$$P(O|P) = P(O|Q) = \prod_{t=1}^{T} P(o_t|q_t)P(q_t|q_{t-1})$$

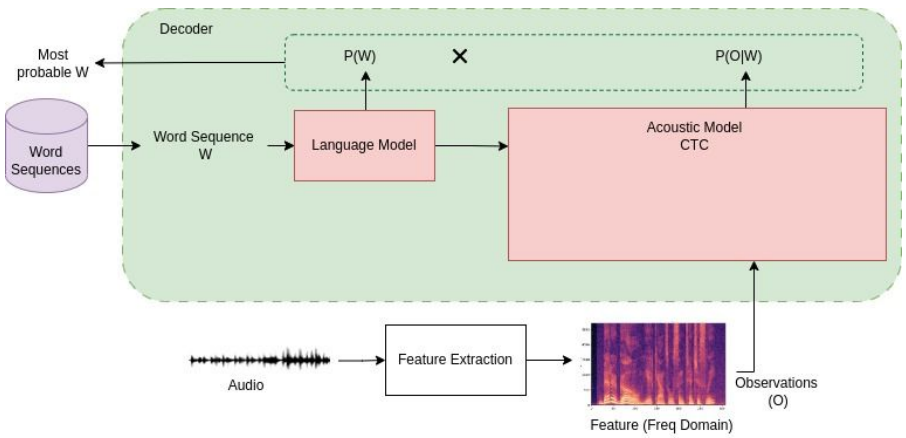


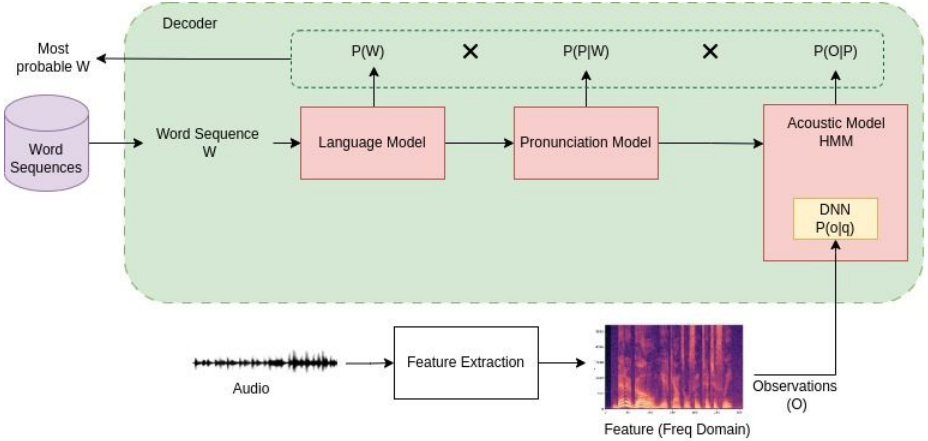
Linear HMM: Start on the left, end on the right.

Source 1

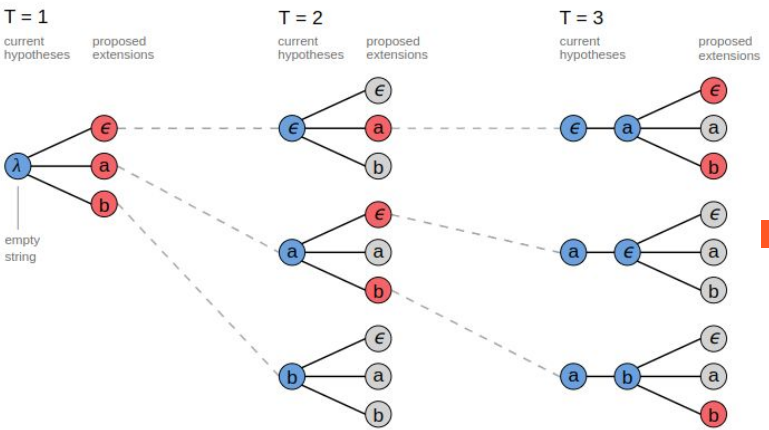
ASR Models - CTC Vs HMM

CTC Vs HMM

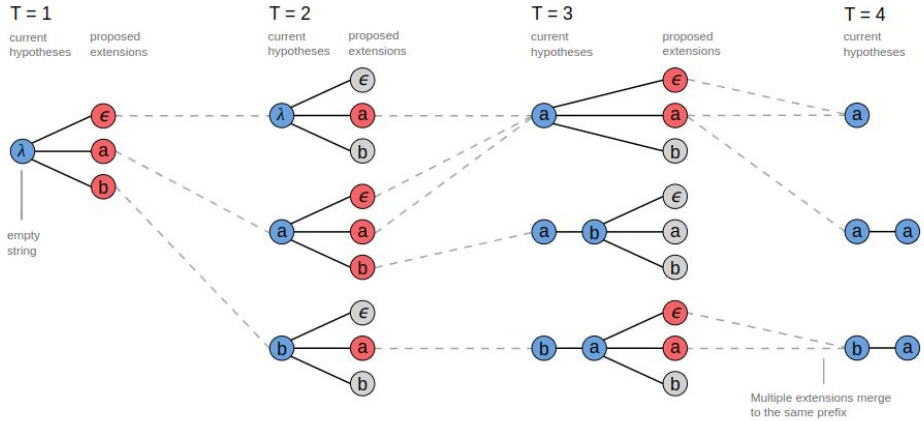




ASR Models - CTC Vs HMM - Beam Search



A standard beam search algorithm with an alphabet of $\{\epsilon, a, b\}$ and a beam size of three.



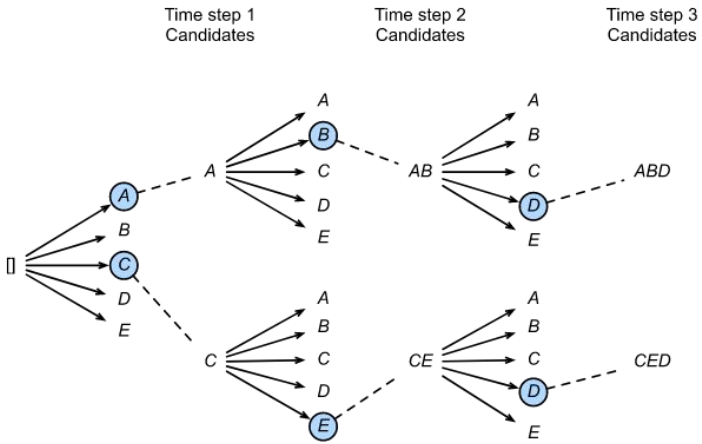
The CTC beam search algorithm with an output alphabet $\{\epsilon, a, b\}$ and a beam size of three.

Source 1

ASR Models - CTC: Inference

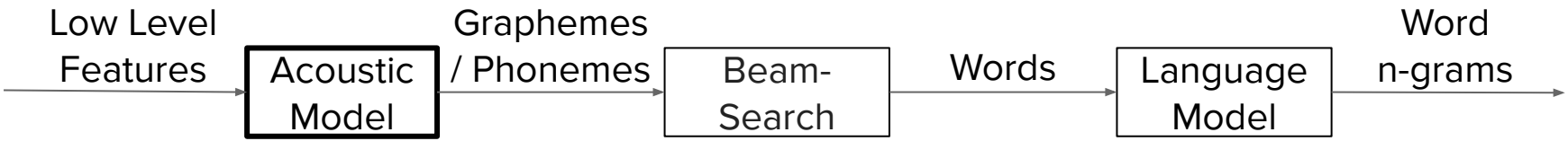
A word on the search

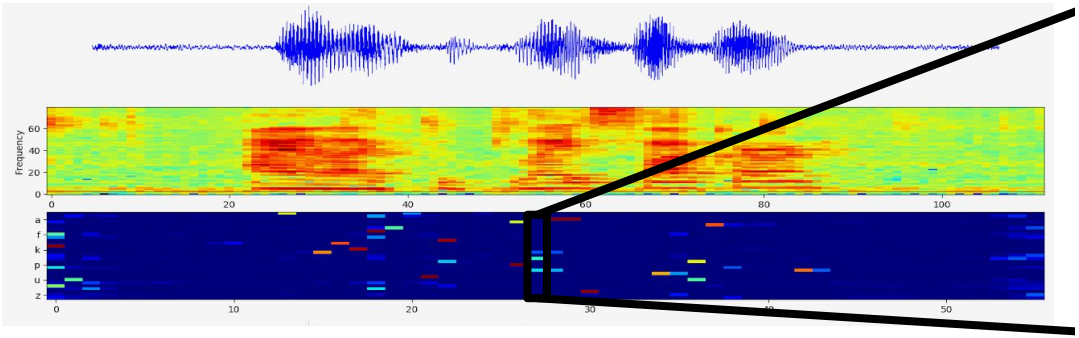
- In practice, it's not practical to search over all word sequences. The search space is too large and exhaustive.
- Solution: beam search

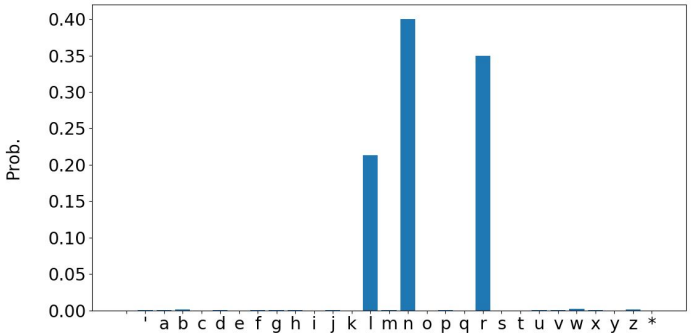


Source 1

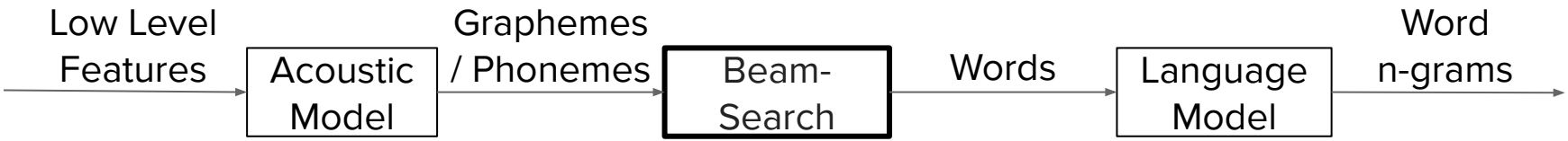
ASR Models - CTC: Inference



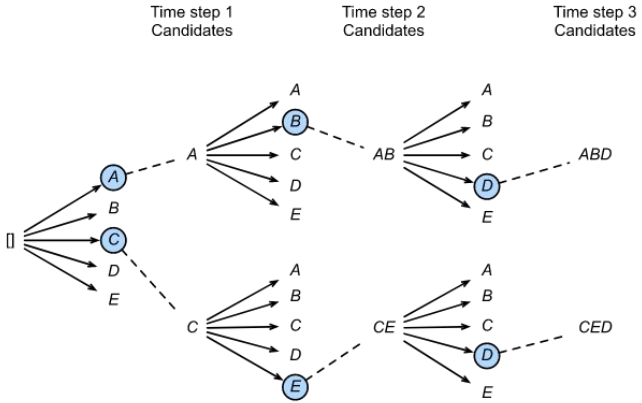




ASR Models - CTC: Inference

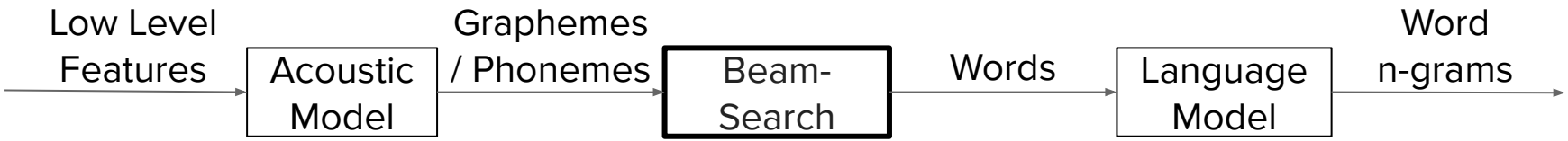


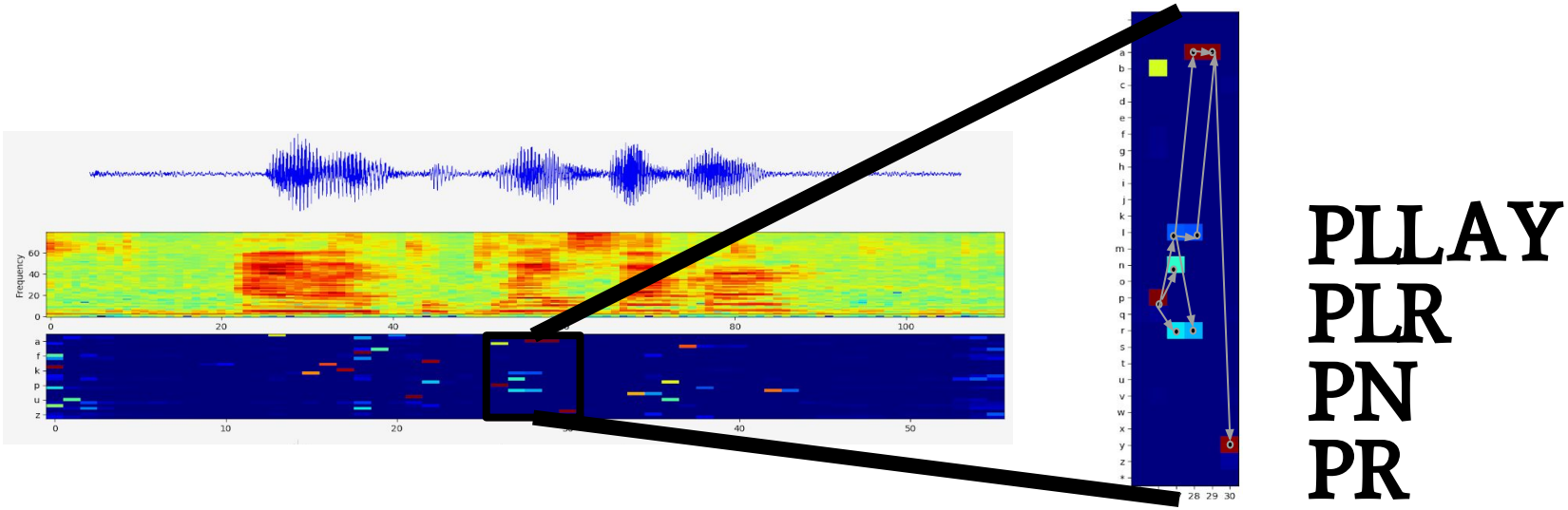
Beam-Search is a (breadth-first) heuristic **search** algorithm that explores a graph by **expanding the most promising nodes in a limited set**



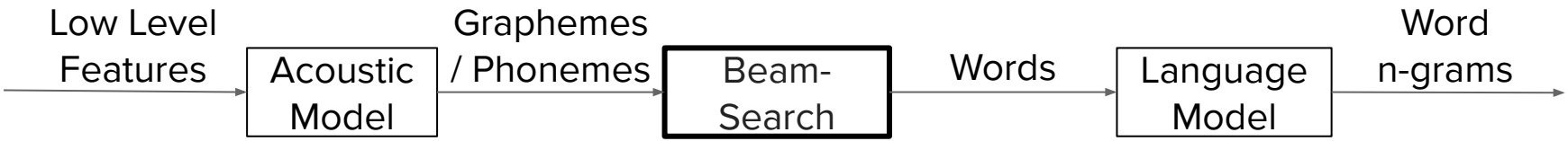
Source 1

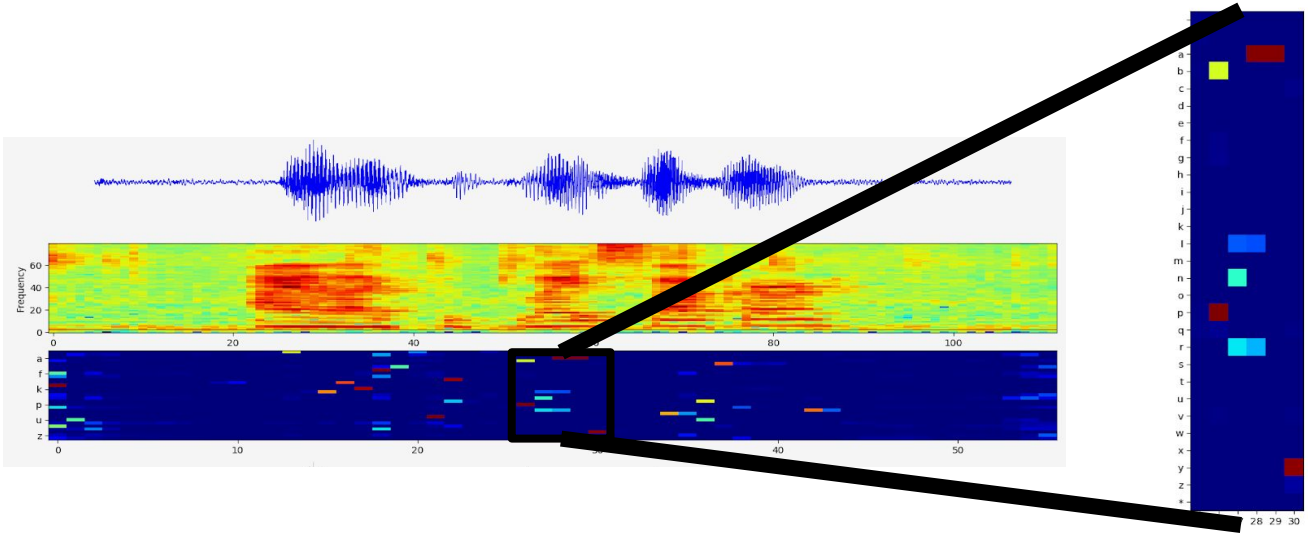
ASR Models - CTC: Inference





ASR Models - CTC: Inference





Beam

play

pray

pnay

----

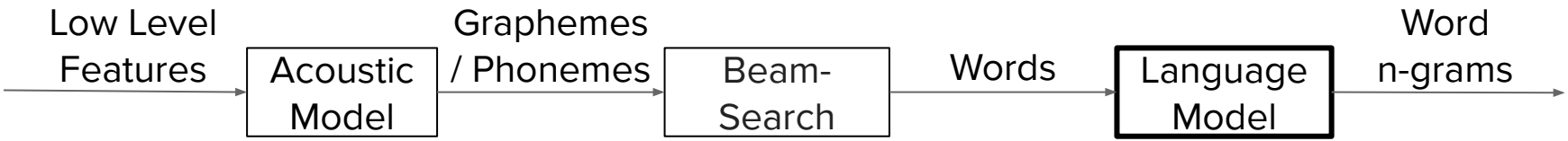
bray

plray

bnay

blay

ASR Models - CTC: Inference



Language model rescores the word sequences and returns the most probable text

I like to play soccer > I like to pray soccer

P("I like to play soccer") = P("I like to") * P("like to play") * P("to play soccer") = 0.9 * 0.8 * 0.85 = **0.612**

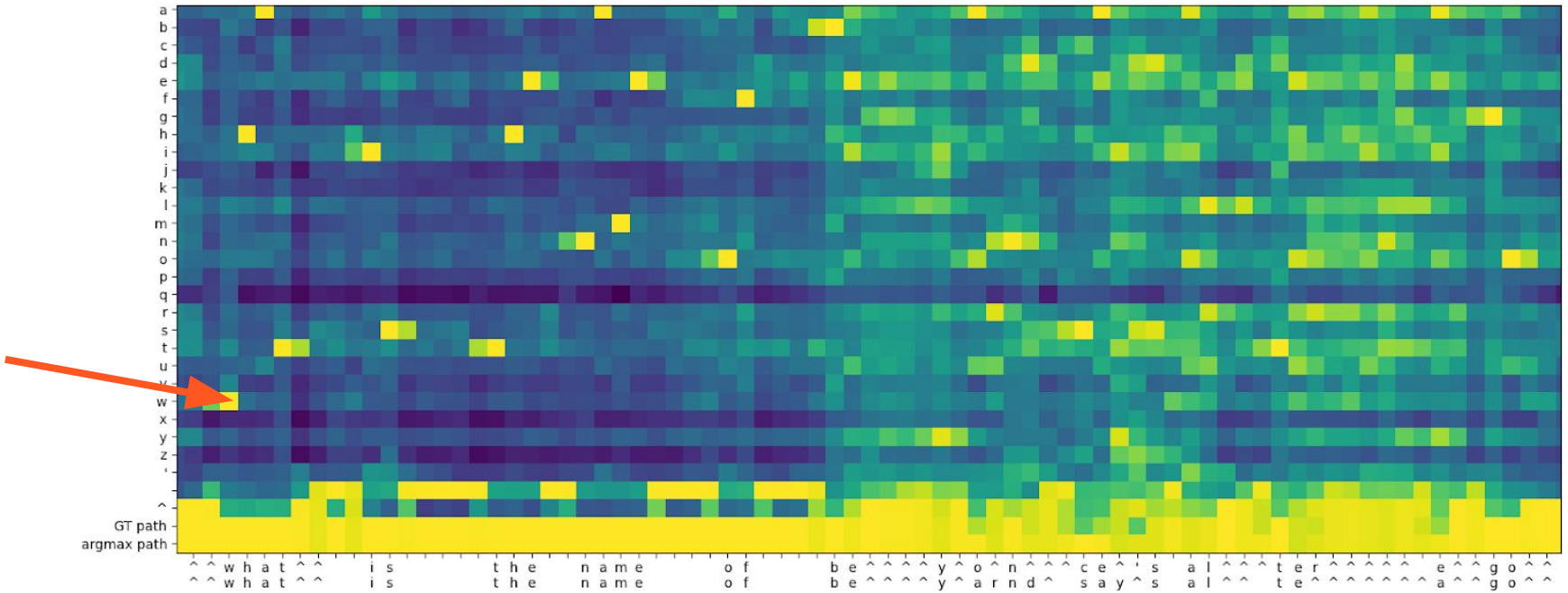
P("I like to pray soccer") = P("I like to") * P("like to pray") * P("to pray soccer") = 0.9 * 0.85 * 0.01 = **0.008**

ASR Models - CTC: Inference

Important Notes

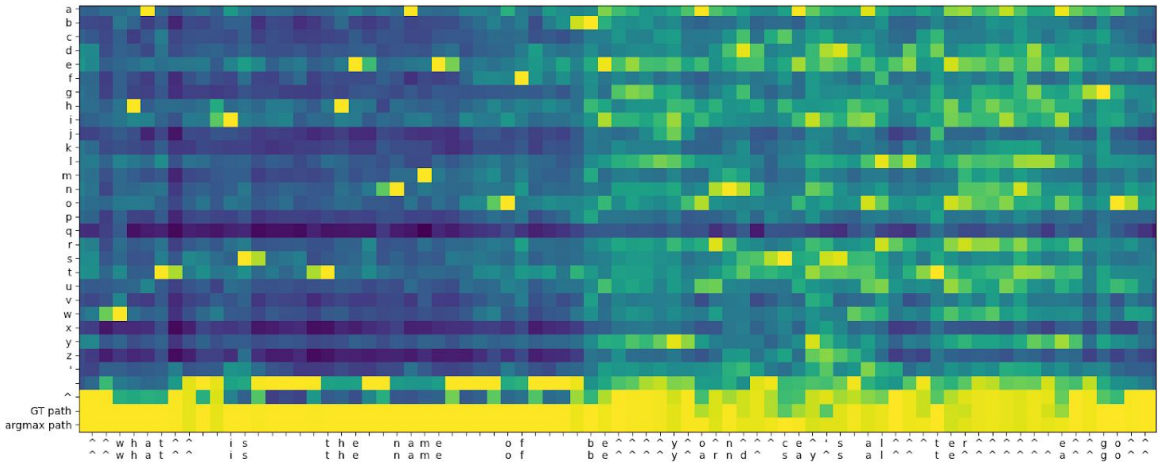
- **Deep Speech**: Scaling up end-to-end speech recognition (Baidu 2014)
- **Optimizing WER instead of CER**: Towards End-to-End Speech Recognition with Recurrent Neural Networks (Graves & Jaitly ICML 2014)
- **Optimizing semantic error rate rather than CES / WER**
- **Acoustic Model Over-Confidence**: Uniform prob. Regularization

ASR Models - CTC: Inference: Example



ASR Models - CTC: Inference: Example

$$argmax_y Pr(y|x) = B(argmax_\pi Pr(\pi|x))$$



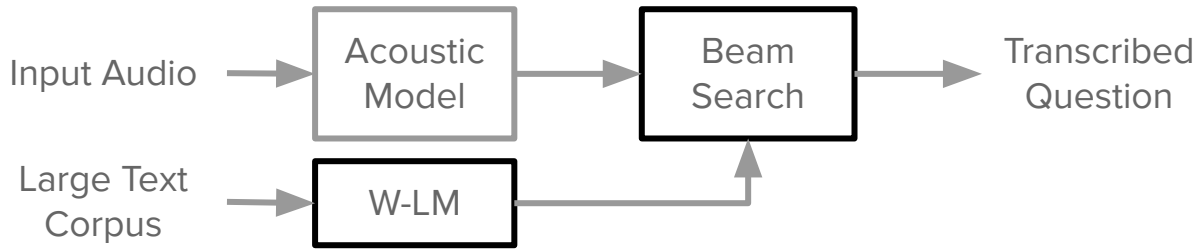
Timed Argmax: ^^what^^ is the name of be^^^^y^arnd^ say^s al^^ te^^^^^^^a^^go^^

Collapsed Argmax: what is the name of beyarnd says al teago

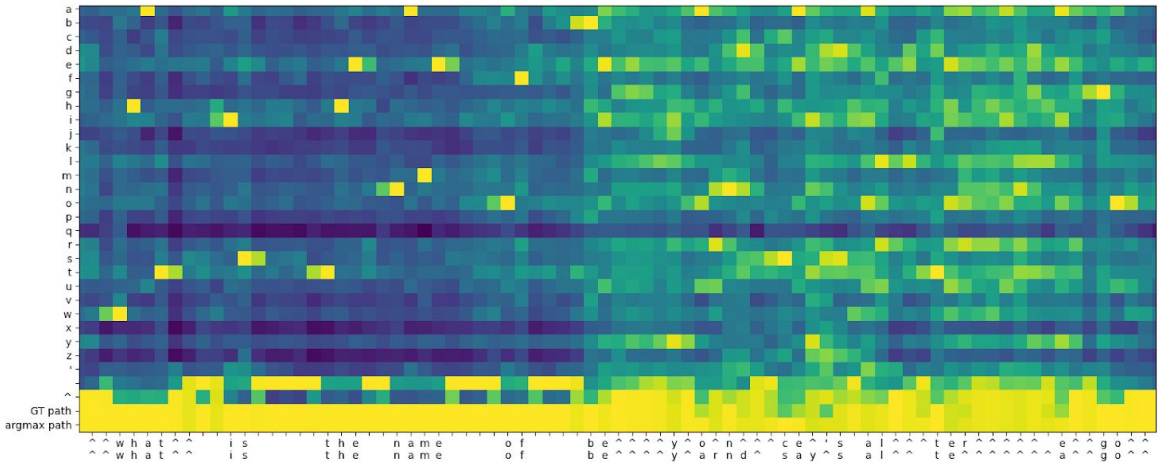
ASR Models - CTC: Inference: Example

Beam-Search is a **heuristic** search algorithm that explores a **graph** by **expanding** the **most promising nodes** in a limited set

$$Pr(y|x) = log(Pr_{AM}(y|x)) + \lambda_W * log(Pr_{W-LM}(y))$$



ASR Models - CTC: Inference: Example

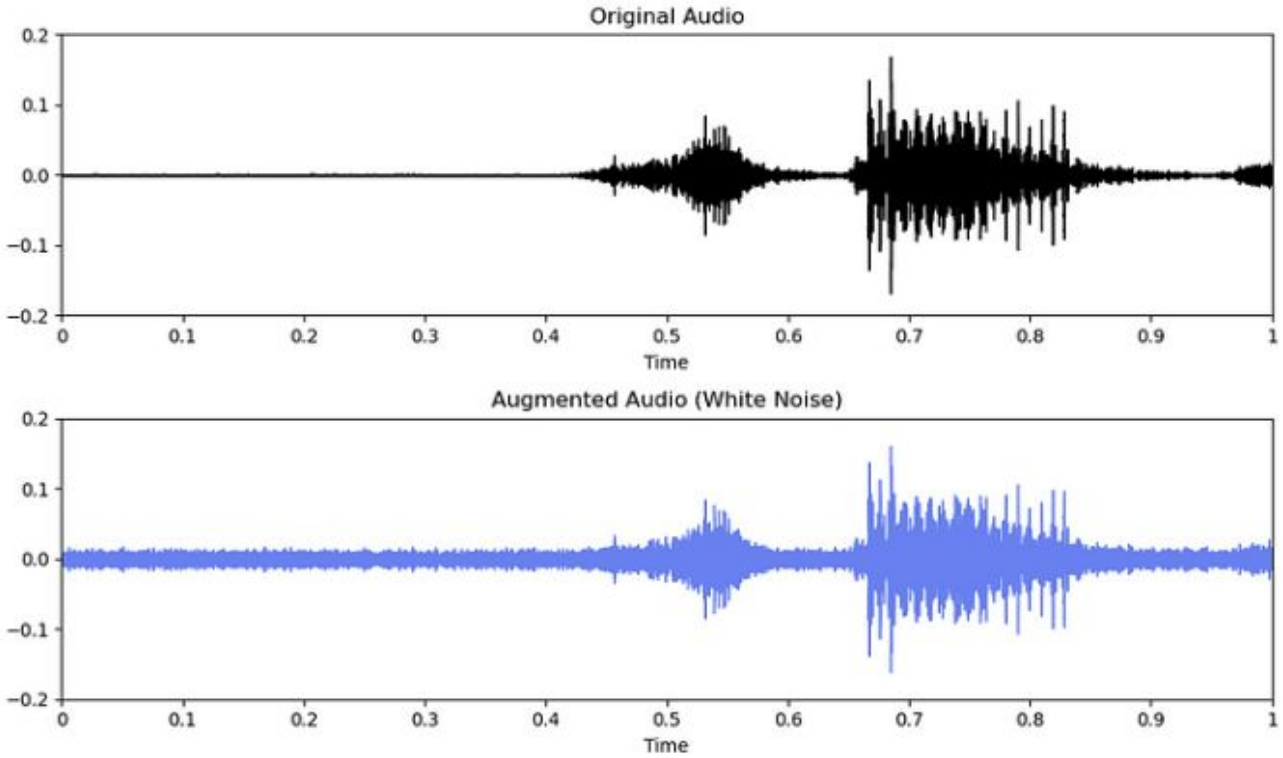


Timed Argmax: ^^what^^ is the name of be^^^^y^arnd^ say^s al^^ te^^^^^^^a^^go^^

Collapsed Argmax: what is the name of beyarnd says al teago

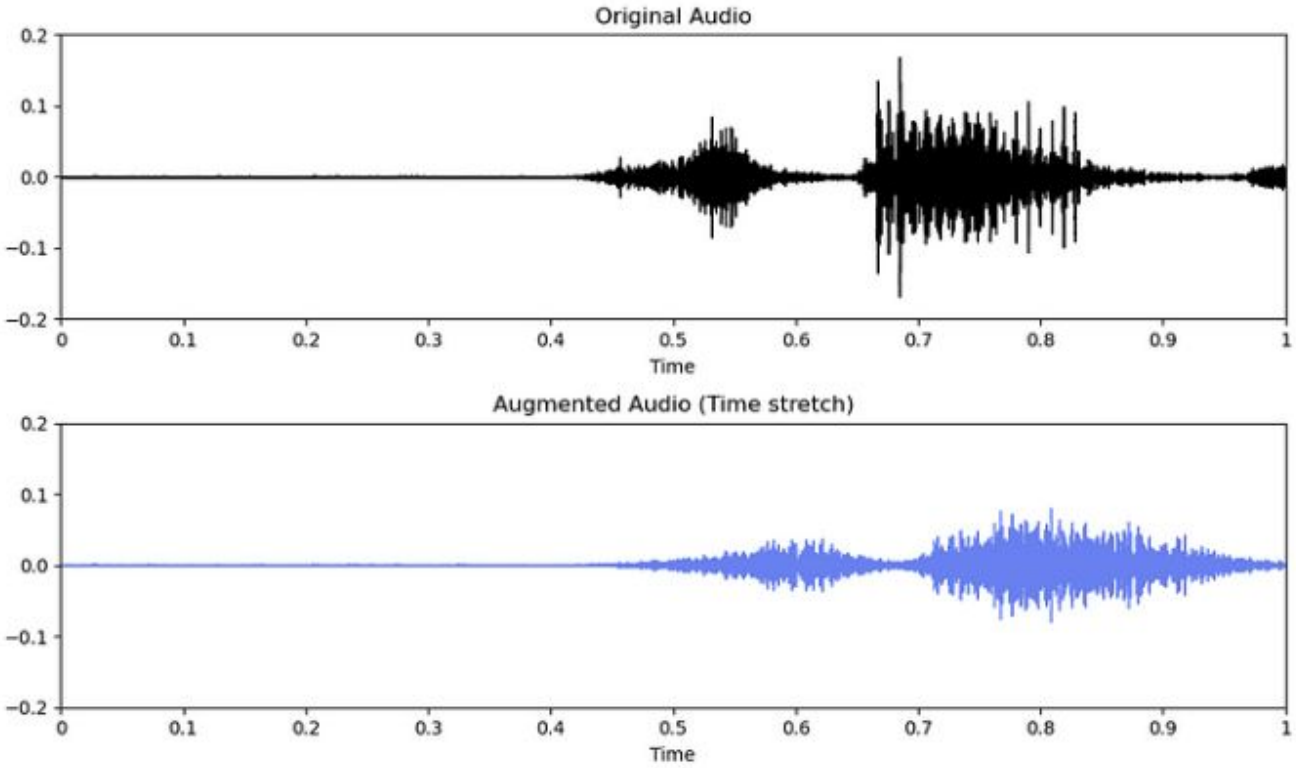
BS + W-LM: what is the name of beyonce's alter ego

ASR Models - CTC: Audio Augmentation



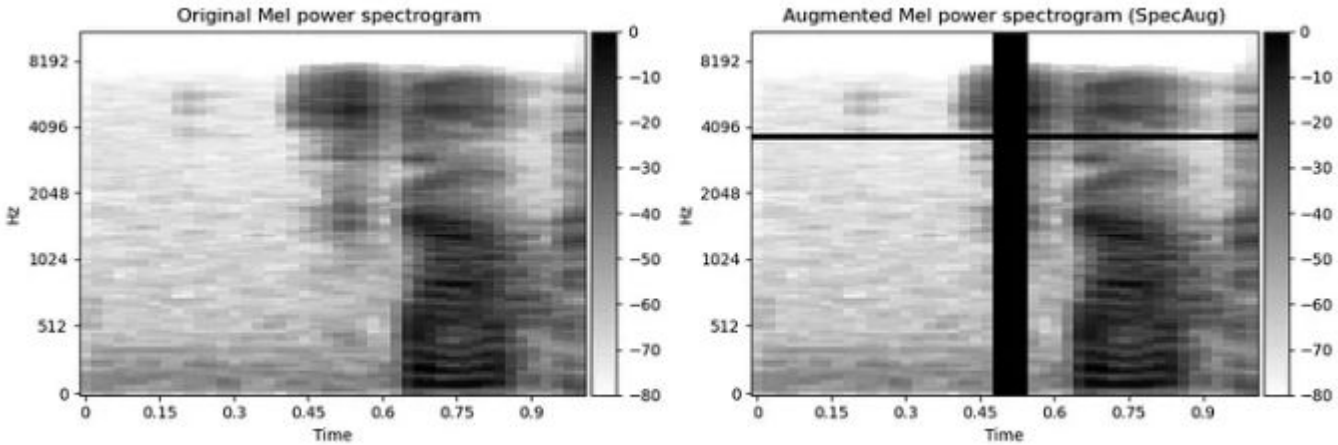
Source 1

ASR Models - CTC: Audio Augmentation



Source 1

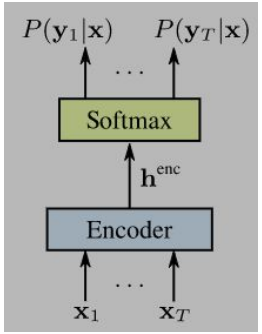
ASR Models - CTC: Audio Augmentation



Source 1

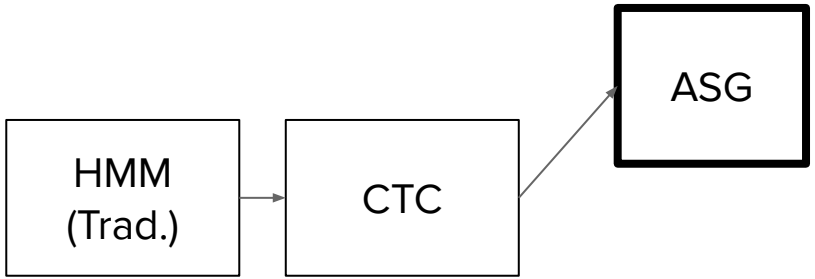
ASR Models - CTC

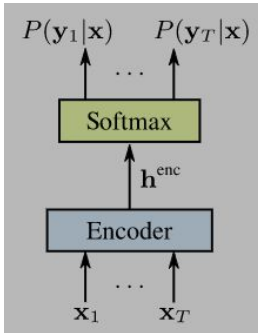
HMM (Trad.) → CTC



A Comparison of Sequence-to-Sequence Models for Speech Recognition. IntraSpeech 2017
Exploring Neural Transducers for End-to-End Speech Recognition (DeepSpeech 3) 2017

ASR Models - ASG



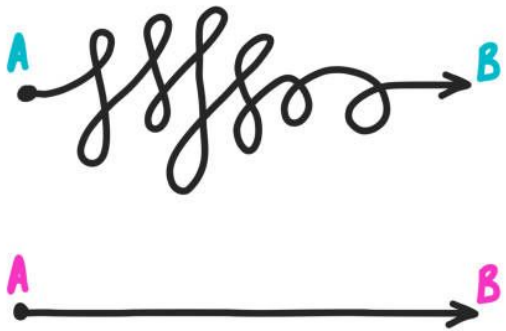


A Comparison of Sequence-to-Sequence Models for Speech Recognition. IntraSpeech 2017
Exploring Neural Transducers for End-to-End Speech Recognition (DeepSpeech 3) 2017

ASR Models - ASG

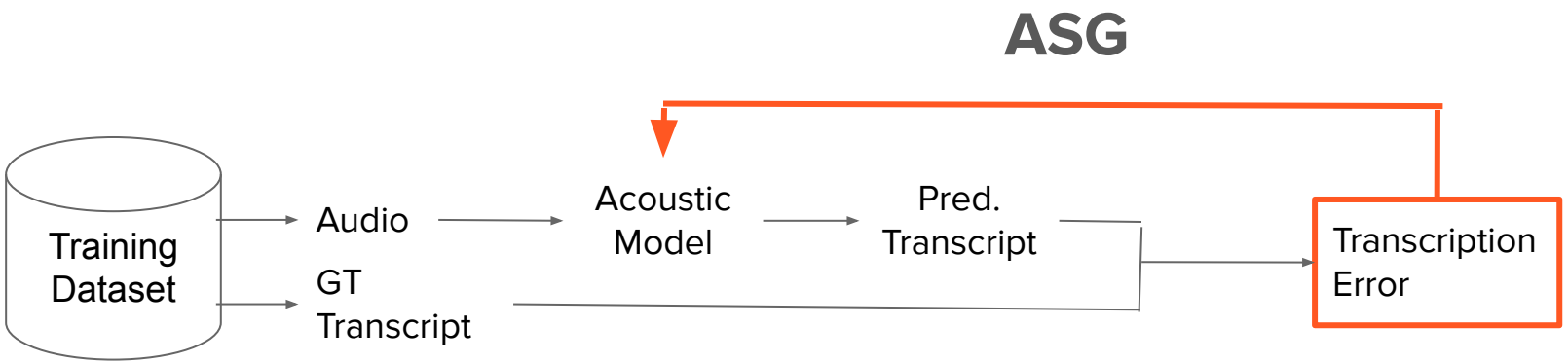
Wav2Letter: an End-to-End ConvNet-based Speech Recognition System (FaceBook 2016)

Auto Segmentation Criterion (ASG) is on-par with CTC while simplifying the decoding process and allowing to integrate the transition probabilities during training.



Source 1

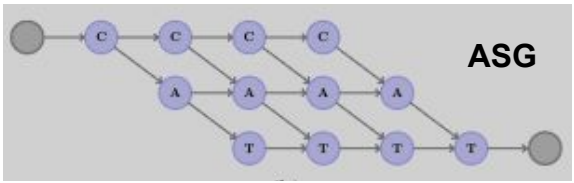
ASR Models - ASG

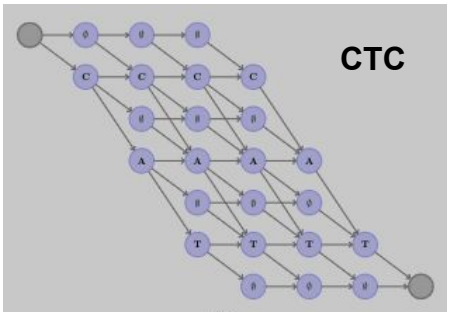


ASR Models - ASG

Main differences from CTC:

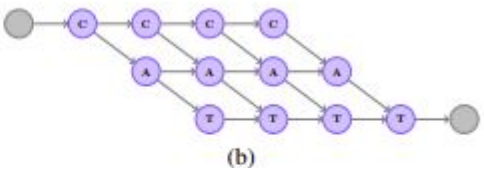
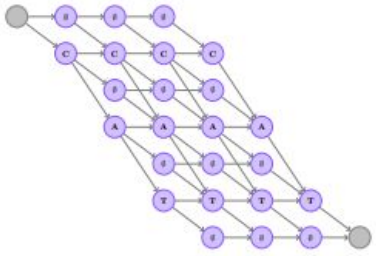
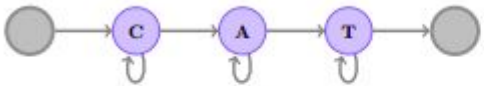
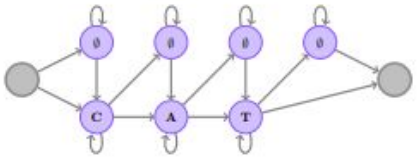
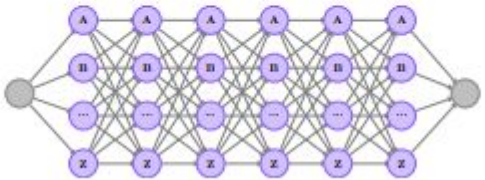
- No blank labels: **Much simpler graph & decoder**
- Adds ‘repetition character label’, e.g. caterpillar -> caterpil2ar
- Learns the **transition probabilities** (using the normalization graph)





Wav2Letter: an End-to-End ConvNet-based Speech Recognition System (FaceBook 2016)

ASR Models - ASG

	ASG	CTC
Unfolded Graph	 (b)	 (b)
Folded Graph	 (a)	 (a)
Normalization Graph		

Wav2Letter: an End-to-End ConvNet-based Speech Recognition System (FaceBook 2016)

ASR Models - ASG

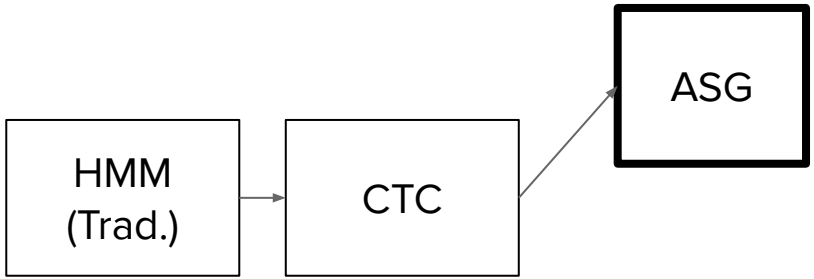
$$ASG(\theta, T) = -\underset{\pi \in \mathcal{G}_{asg}(\theta,T)}{\text{logadd}} \sum_{t=1}^{T} (f_{\pi_t}(x) + g_{\pi_{t-1},\pi_t}(x)) + \underset{\pi \in \mathcal{G}_{full}(\theta,T)}{\text{logadd}} \sum_{t=1}^{T} (f_{\pi_t}(x) + g_{\pi_{t-1},\pi_t}(x))$$

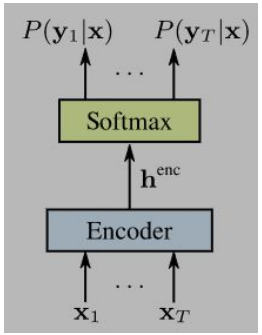
Acoustic Model's Outputs (No-Softmax)

Transition Scores

Wav2Letter: an End-to-End ConvNet-based Speech Recognition System (FaceBook 2016)

ASR Models - ASG

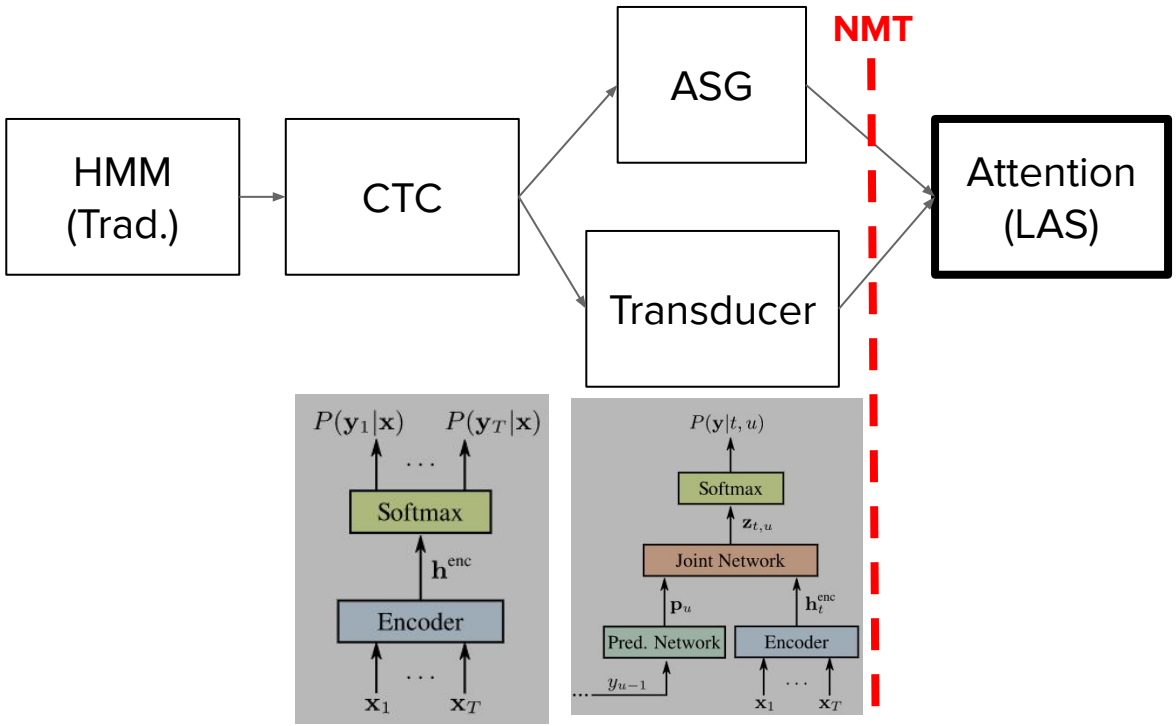




A Comparison of Sequence-to-Sequence Models for Speech Recognition. IntraSpeech 2017

Exploring Neural Transducers for End-to-End Speech Recognition (DeepSpeech 3) 2017

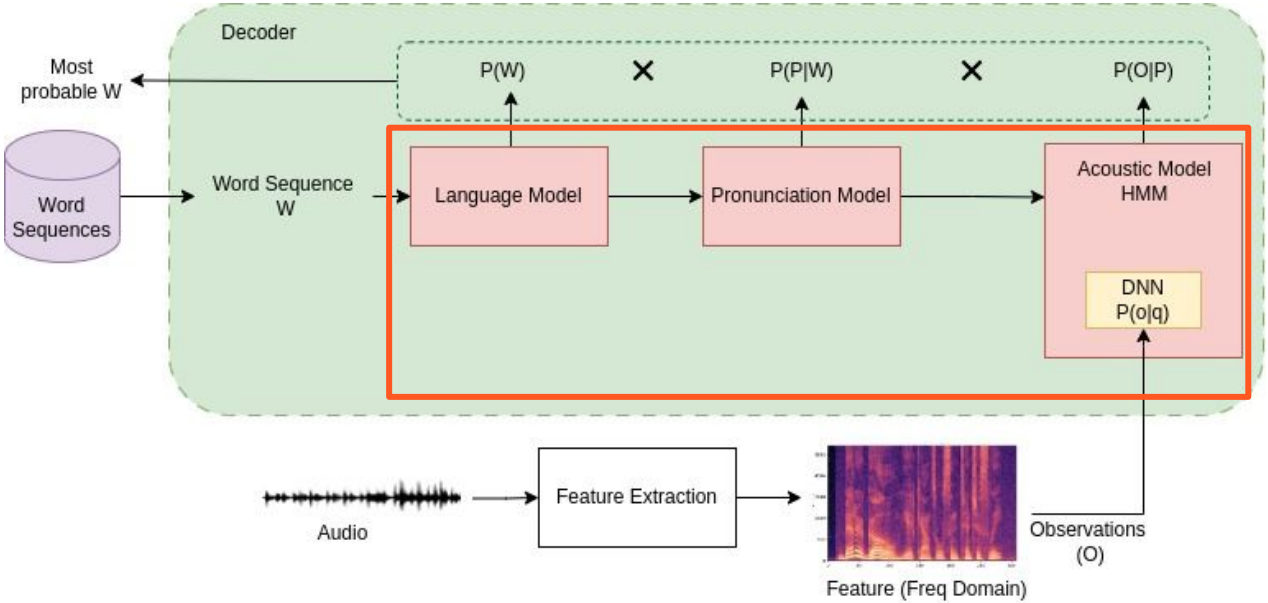
ASR Models - LAS



A Comparison of Sequence-to-Sequence Models for Speech Recognition. IntraSpeech 2017

Exploring Neural Transducers for End-to-End Speech Recognition (DeepSpeech 3) 2017

ASR Models - LAS



Source 1, 2

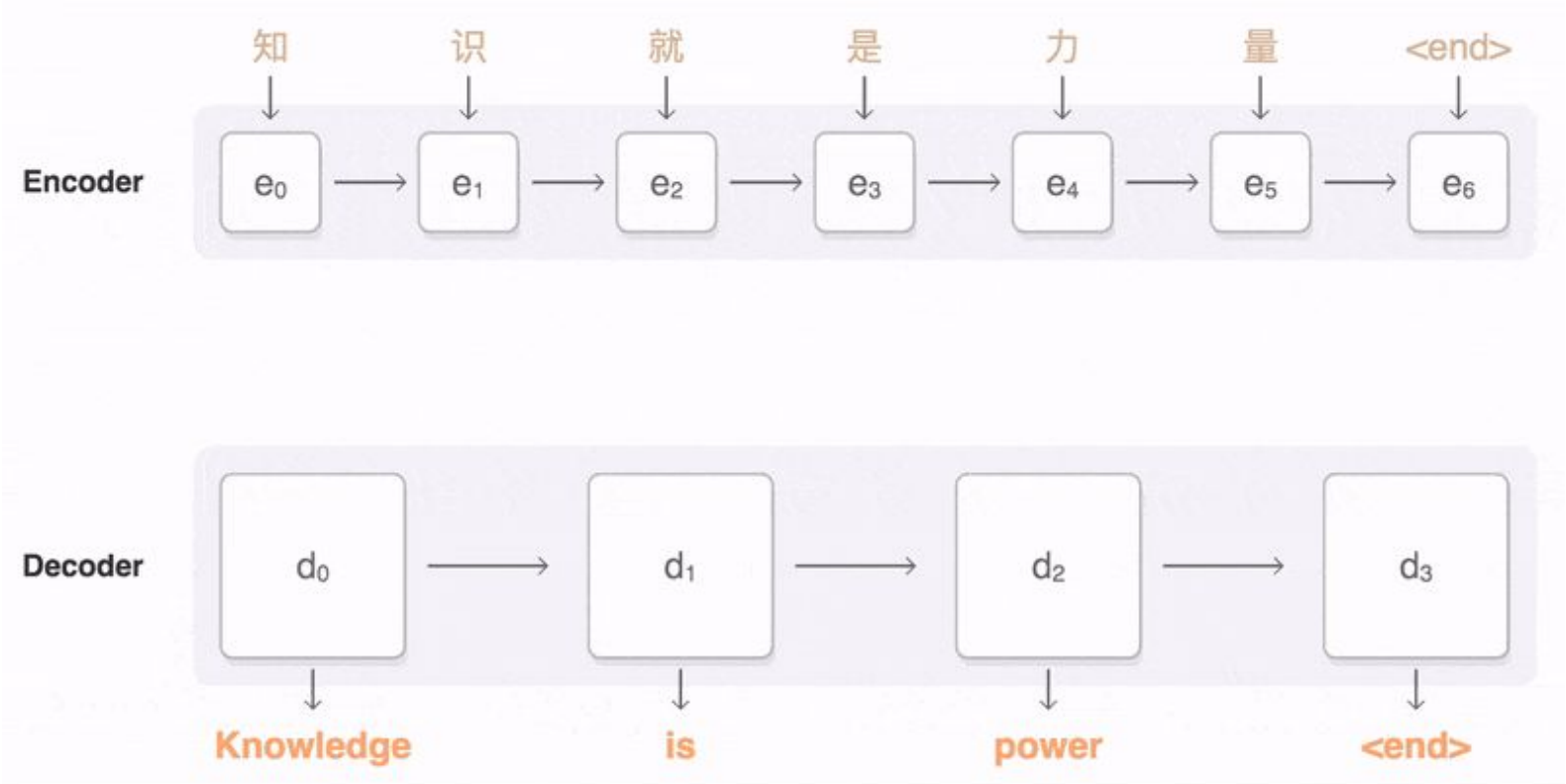
ASR Models - LAS

Listen Attend Spell (LAS), William Chan, Navdeep Jaitly, Quoc V. Le, Oriol Vinyals (Google), 2015

Jointly trained encoder-decoder (with attention) based model

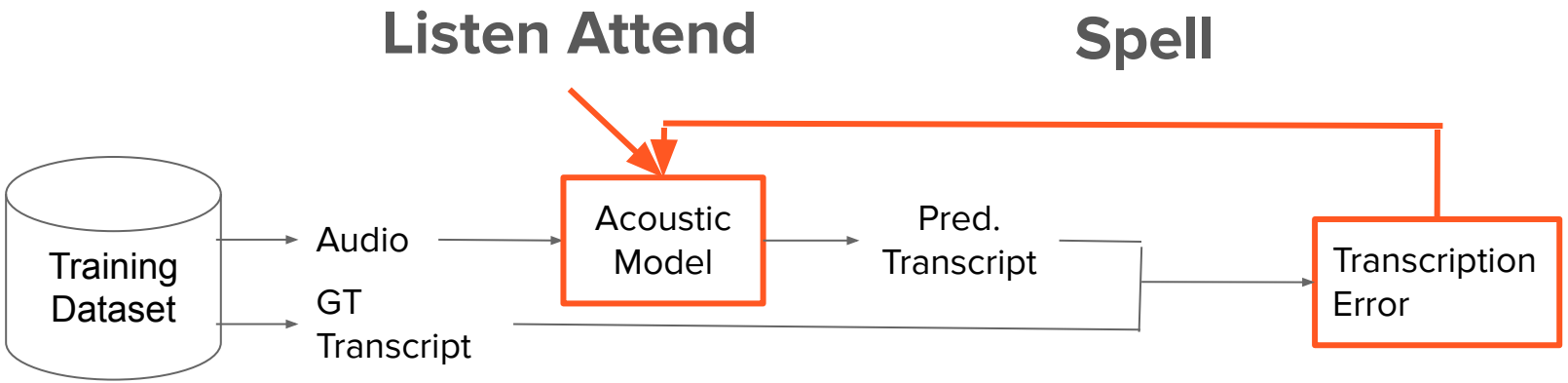
Source 1

ASR Models - LAS



Source 1

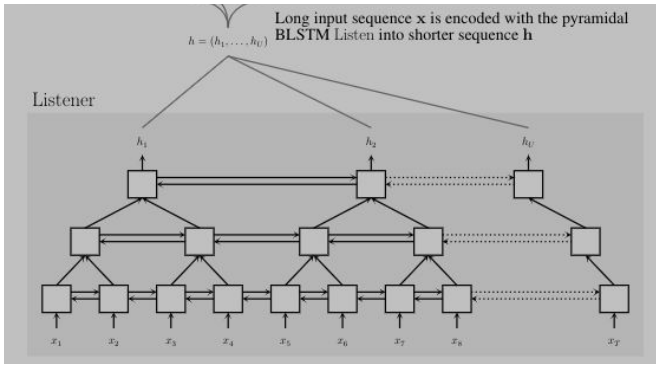
ASR Models - LAS



ASR Models - LAS

Encoder: **Pyramidal** RNN That **Converts Low Level** Features **Into High Level** Features (Could be replaced with any other encoder)

$$\mathbf{h} = Listen(\boldsymbol{x})$$



Listen Attend Spell, William Chan, Navdeep Jaitly, Quoc V. Le, Oriol Vinyals (Google), 2015

ASR Models - LAS

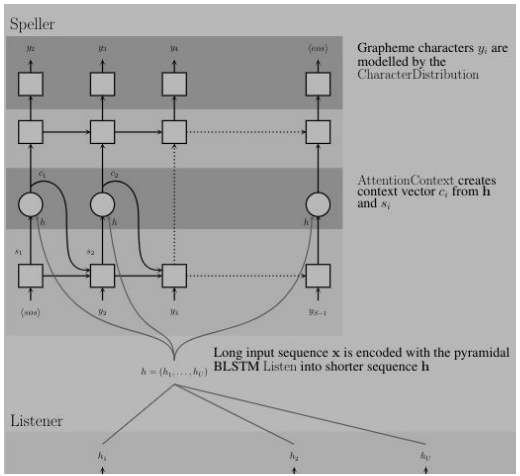
Decoder: **Attention-Based** Recurrent Network That **Converts** These High Level Features **Into Text**, **One Character At A Time**.

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

LAS Maximizes (NO MARGINALIZATION):

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

Removes Independence Assumption Between Outputs

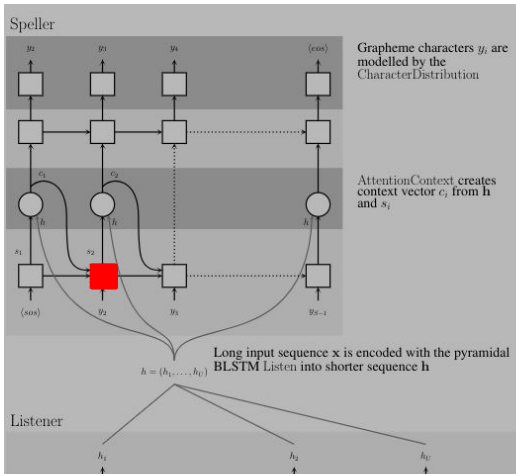


Listen Attend Spell, William Chan, Navdeep Jaitly, Quoc V. Le, Oriol Vinyals (Google), 2015

ASR Models - LAS

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

$$s_i = RNN(s_{i-1}, y_{i-1}, c_{i-1})$$

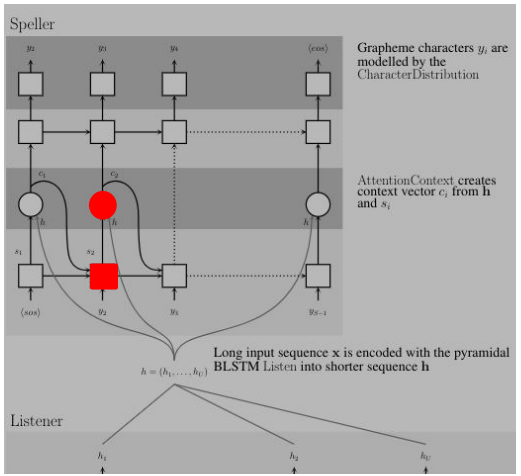


ASR Models - LAS

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

$$s_i = RNN(s_{i-1}, y_{i-1}, c_{i-1})$$

$$c_i = AttentionContext(s_i, h)$$



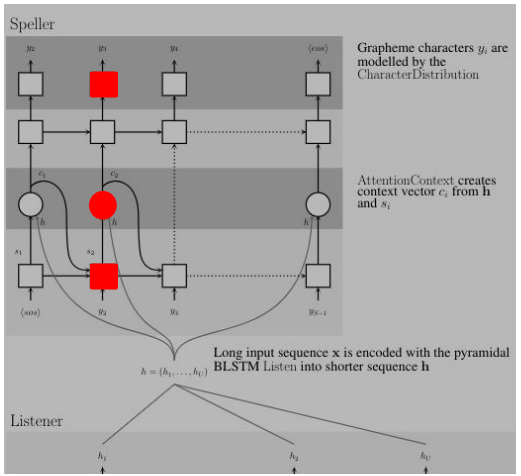
ASR Models - LAS

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

$$s_i = RNN(s_{i-1}, y_{i-1}, c_{i-1})$$

$$c_i = AttentionContext(s_i, h)$$

$$P(y_i|x, y_{<i}) = CharacterDistribution(s_i, c_i)$$



ASR Models - LAS

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

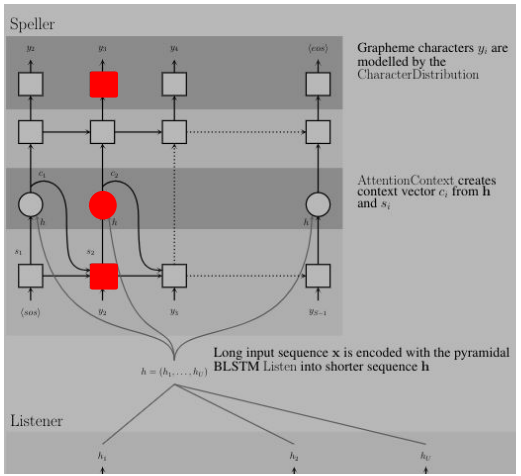
$$s_i = RNN(s_{i-1}, y_{i-1}, c_{i-1})$$

$$c_i = AttentionContext(s_i, h)$$

$$P(y_i|x, y_{<i}) = CharacterDistribution(s_i, c_i)$$

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

Introducing **<SOS>** and **<EOS>** tokens.



ASR Models - LAS

Two (2) steps attention mechanism: Neural Machine Translation by Jointly Learning to Align and Translate, Bahadanau et al. 2014.

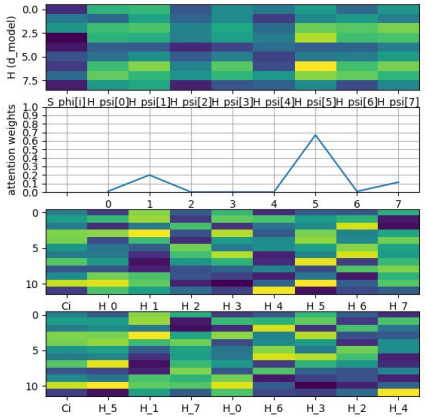
$$c_i = AttentionContext(s_i, h)$$

$$e_{i,u} = \langle \phi(s_i), \psi(h_u) \rangle$$ Dot Product

$$a_{i,u} = \frac{exp(e_{i,u})}{\sum_u exp(e_{i,u})}$$ Probs. (Attention)

$$c_i = \sum_u a_{i,u} h_u$$ Context Vector

```
T=8, d_model=9
S.shape=(10, 8), H.shape=(12, 8)
S_phi.shape=(9, 8), H_psi.shape=(9, 8)
S_phi_i.shape=(9, 1)
e_i.shape=(8,)
alpha_i.shape=(8,)
c_i.shape=(12,)
```



ASR Models - LAS

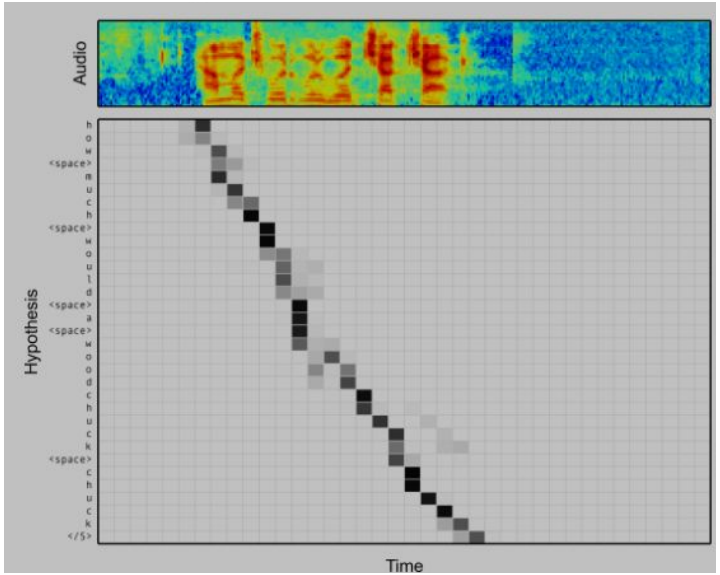
Two (2) steps attention mechanism: Neural Machine Translation by Jointly Learning to Align and Translate, Bahadanau et al. 2014.

$$c_i = AttentionContext(s_i, h)$$

$$e_{i,u} = \langle \phi(s_i), \psi(h_u) \rangle$$ Dot Product

$$a_{i,u} = \frac{exp(e_{i,u})}{\sum_u exp(e_{i,u})}$$ Probs. (Attention)

$$c_i = \sum_u a_{i,u} h_u$$ Context Vector



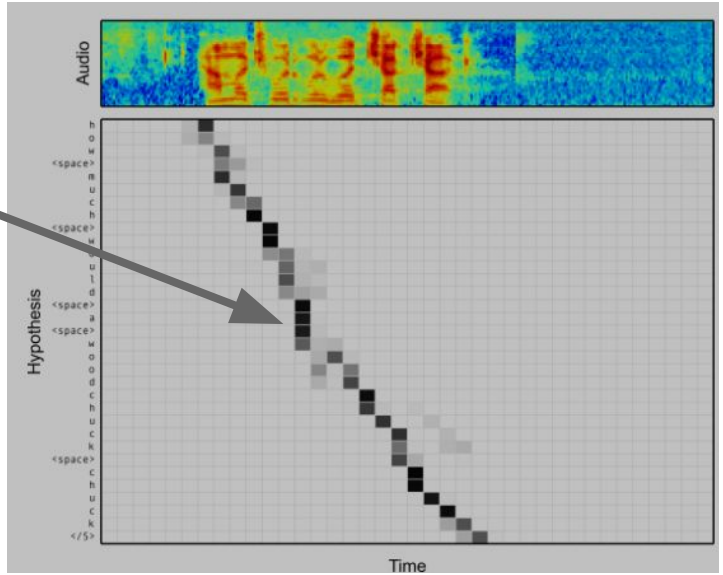
ASR Models - LAS

The **Encoder-Decoder enables**:

- **Emits multiple chars per time-step**
- **Saves computations** (using Encoder's max-pool layers)
- **Enables larger context**, e.g. 1988989898989

The **Attention mechanism** enables:

- Serves as a **skip connection**
- Solves the **alignment**



ASR Models - LAS

Train: LAS maximizes:

$$max_\theta \sum_i log(P(y_i|x, y^*_{<i}; \theta))$$

Where $y^*_{<i}$ is the **ground truth** of the previous character

Train / Inference Incompatibility: Use 10% of the times the previous step prediction instead of the GT

Acoustic Model's Overconfidence: Label smoothing

ASR Models - LAS

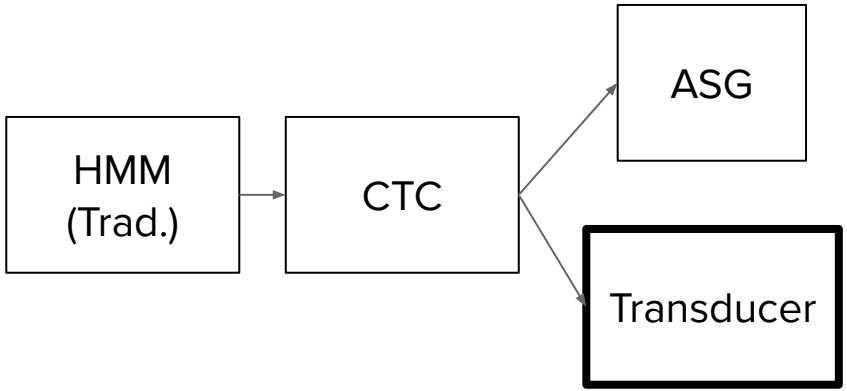
Left-To-Right Beam Search

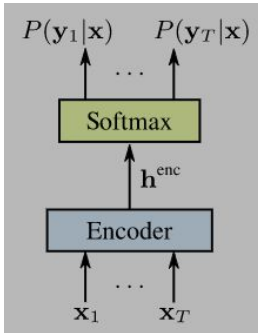
ASR Models - LAS

Notes:

- **Rare & OOV** words are **handled automatically:** Model outputs the character sequence, one character at a time.
- **No conditional independence assumption** enables to generate **several labels** at a time:
  - “triple a” - “triple a” and “aaa” in the top beams.
  - CTC will have problem in such cases.

ASR Models - RNN-T





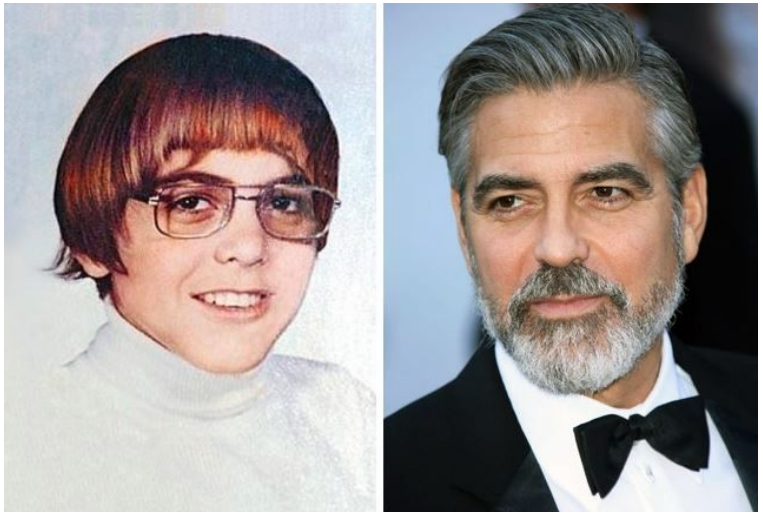
A Comparison of Sequence-to-Sequence Models for Speech Recognition. IntraSpeech 2017

Exploring Neural Transducers for End-to-End Speech Recognition (DeepSpeech 3) 2017

ASR Models - RNN-T

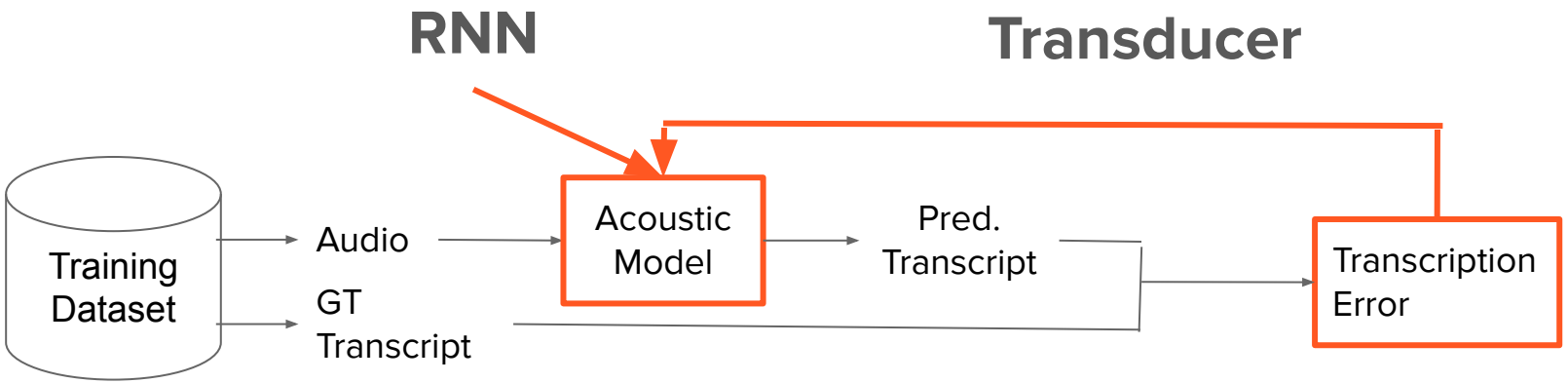
Sequence Transduction with Recurrent Neural Networks, Alex Graves, 2012

“Late Bloomer”



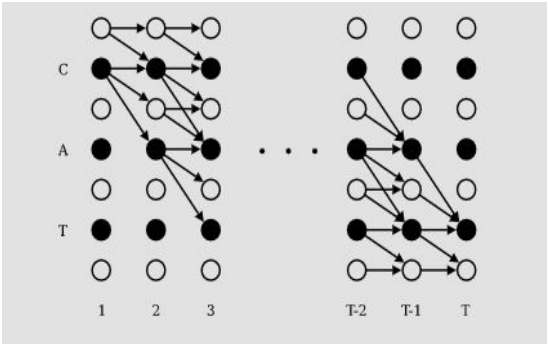
Source 1

ASR Models - RNN-T



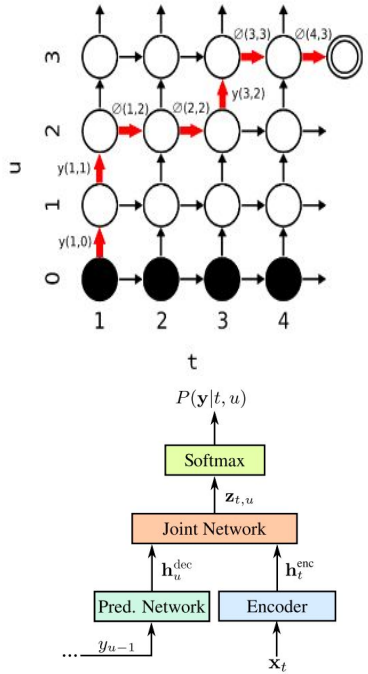
ASR Models - RNN-T

CTC



Towards End-to-End Speech Recognition with Recurrent Neural Networks (Graves & Jaitly 2014)

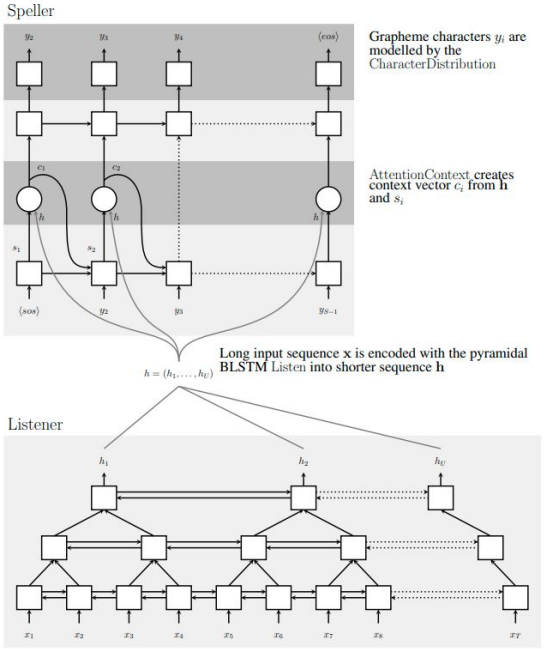
RNN-T



Sequence Transduction with Recurrent Neural Networks, Alex Graves, 2012

https://ai.googleblog.com/2019/03/an-all-neural-on-device-speech.html

LAS (Seq2Seq)



Listen Attend Spell, William Chan, Navdeep Jaitly, Quoc V. Le, Oriol Vinyals (Google), 2015

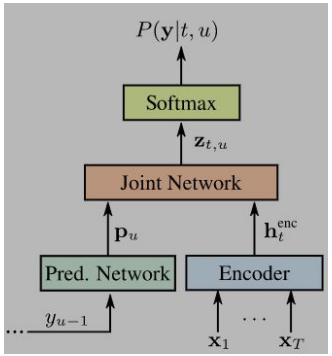
ASR Models - RNN-T

The transducer removes the interdependent assumption between the outputs (unlike CTC)

by defining a distribution over **output sequences of all lengths**, and by jointly modelling both input-output and **output-output dependencies**.

ASR Models - RNN-T

CTC model can be viewed as an **Acoustic Model. The RNN-Transducer augments** the encoder network with a separate **recurrent prediction network** that can be viewed as a **language model**



Sequence Transduction with Recurrent Neural Networks, Alex Graves, 2012

ASR Models - RNN-T

Annotations (same as CTC):

- L : AlphaBet
- $L' = L \bigcup \{blank\}$
- y_k^t : Probability of observing label *k* at time *t*
- $\pi = L'^T$: Path - a sequence of length T

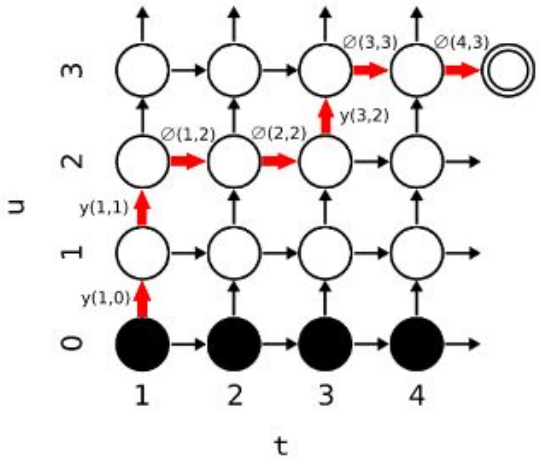
ASR Models - RNN-T

Define many-to-one map: removes only blank labels

$$B : L'^{T} \mapsto L^{\leqslant T}$$
$$B(- - aa - b) = aab$$

RNN-Transducer maximizes:

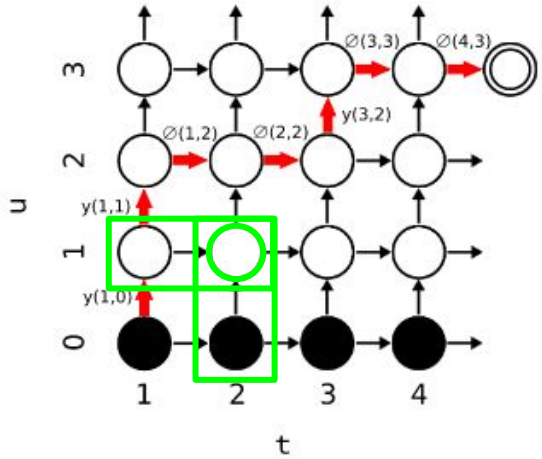
$$p(\mathbf{l}|x) = \sum_{\pi \in B^{-1}(\mathbf{l})} p(\pi|x); \pi \in L'^{\leqslant U+T}$$



Sequence Transduction with Recurrent Neural Networks, Alex Graves, 2012

ASR Models - RNN-T

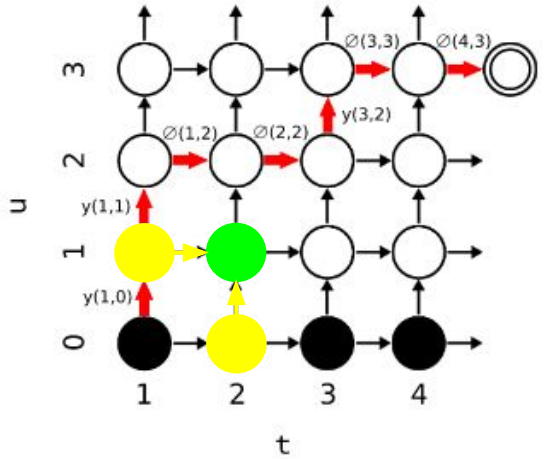
Training using an efficient DP forward-backward algorithm



Sequence Transduction with Recurrent Neural Networks, Alex Graves, 2012

ASR Models - RNN-T

Training using an efficient DP forward-backward algorithm



Sequence Transduction with Recurrent Neural Networks, Alex Graves, 2012

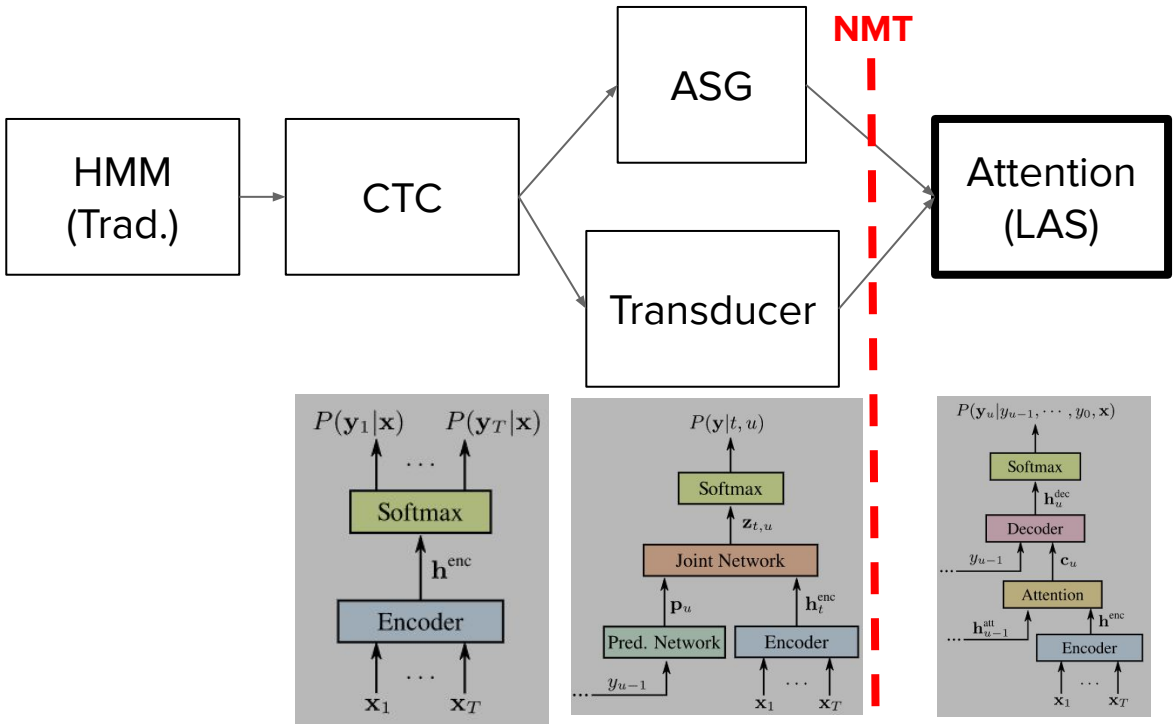
ASR Models - RNN-T

Decoding using:
Naive arg-max (pred. network) / beam-search

ASR Models - RNN-T

- Same as CTC:
  - **Collapse Function:** Due to blank label
  - **Same Optimization Criterion:** Sum all label's paths
  - **Streaming**
- Improvements over CTC:
  - **Removes independent assumption** between outputs
  - Has a 'built-in' Language Model
  - Can **emit more than a single char** per time-step.

ASR Models - Timeline



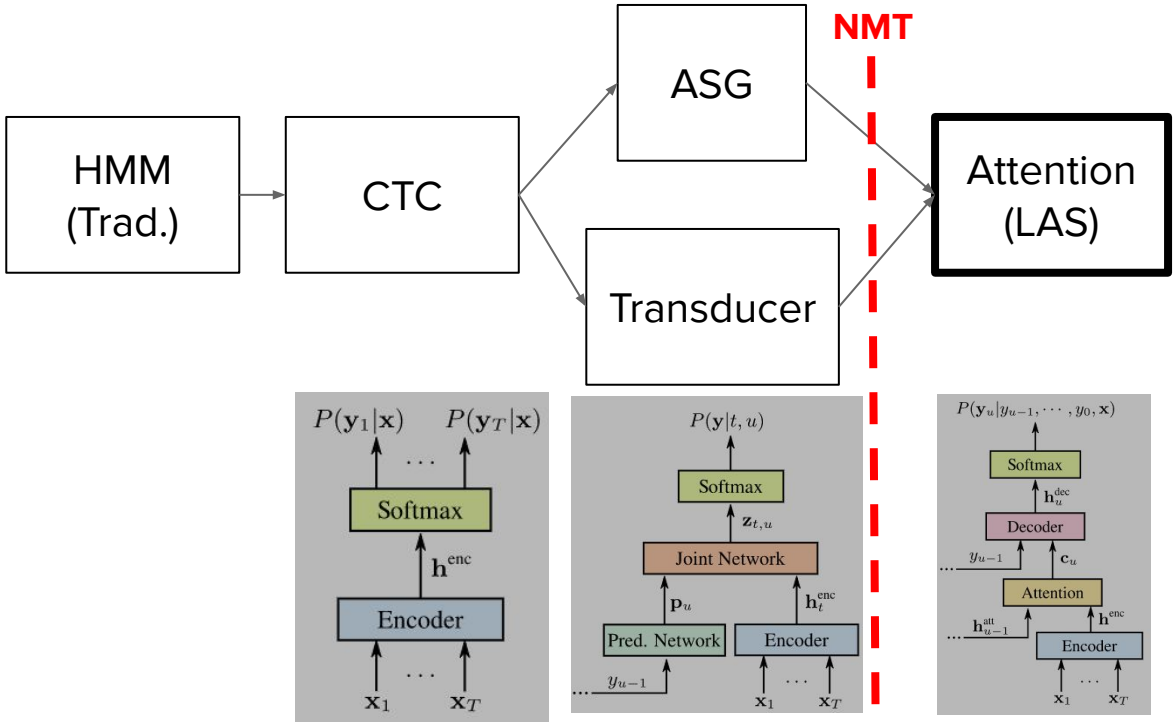
A Comparison of Sequence-to-Sequence Models for Speech Recognition. IntraSpeech 2017

Exploring Neural Transducers for End-to-End Speech Recognition (DeepSpeech 3) 2017

$$p(\mathbf{l}|x) = \sum_{\pi \epsilon B^{-1}(\mathbf{l})} p(\pi|x)$$

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

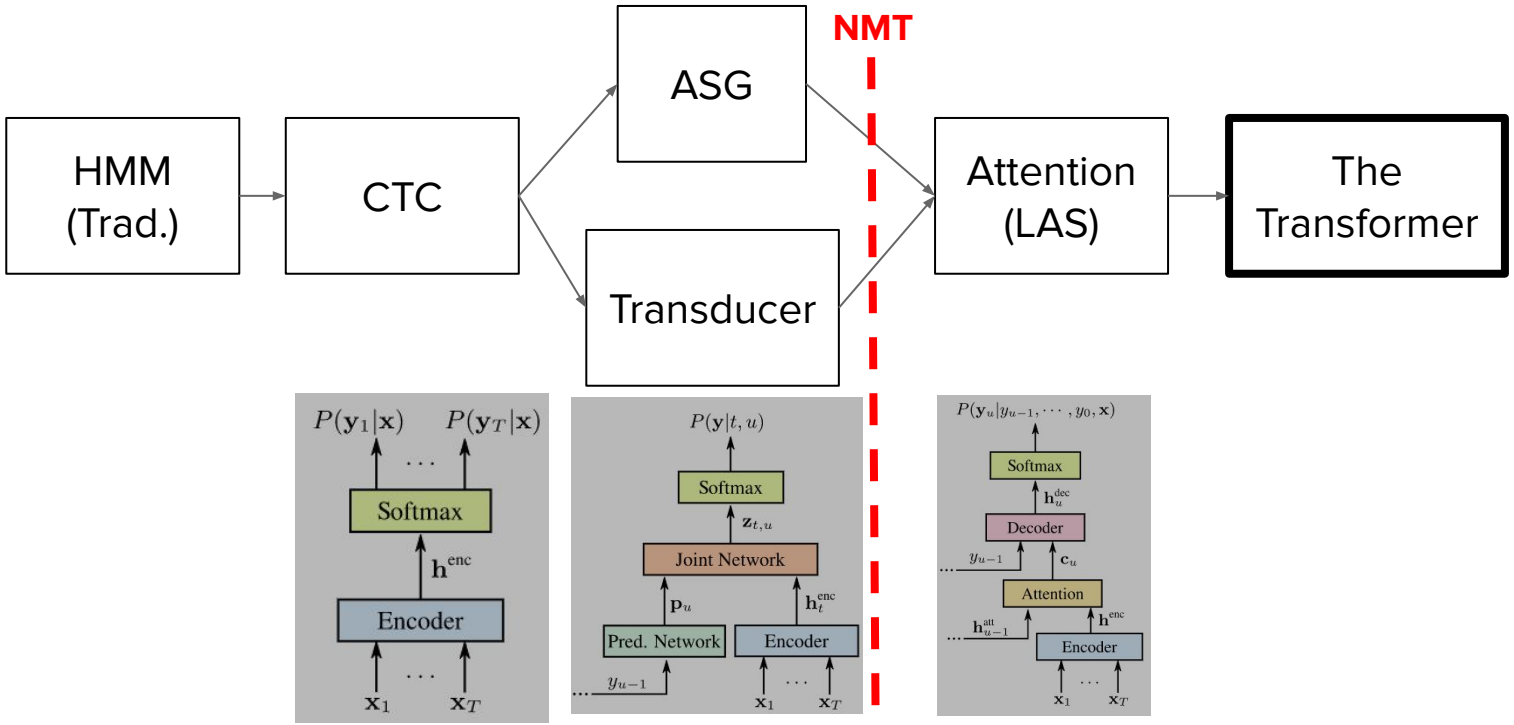
ASR Models - Timeline



A Comparison of Sequence-to-Sequence Models for Speech Recognition. IntraSpeech 2017

Exploring Neural Transducers for End-to-End Speech Recognition (DeepSpeech 3) 2017

ASR Models - Transformer



A Comparison of Sequence-to-Sequence Models for Speech Recognition. IntraSpeech 2017

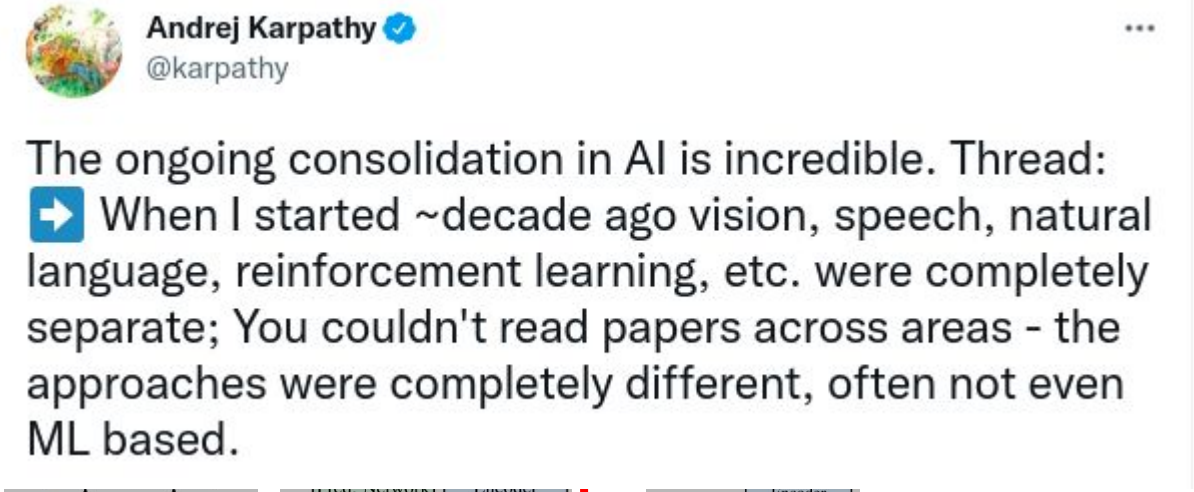
Exploring Neural Transducers for End-to-End Speech Recognition (DeepSpeech 3) 2017

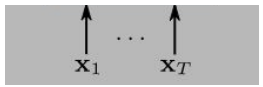
ASR Models - Transformer

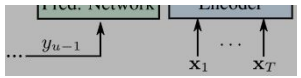
HMM (Trad.)

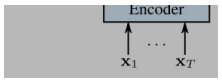
ASG

NMT





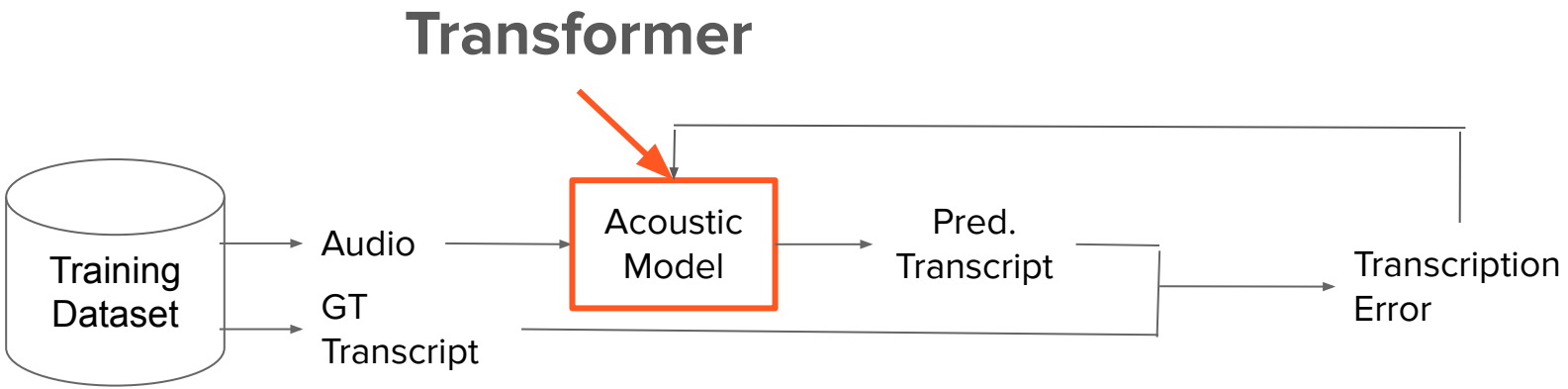




A Comparison of Sequence-to-Sequence Models for Speech Recognition. IntraSpeech 2017

Exploring Neural Transducers for End-to-End Speech Recognition (DeepSpeech 3) 2017

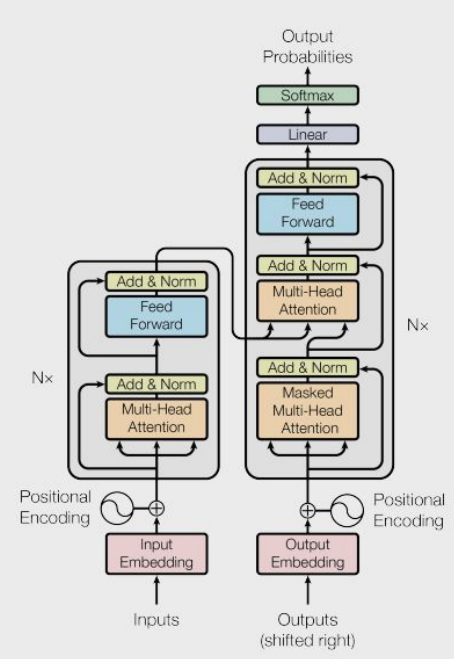
ASR Models - Transformer



ASR Models - Transformer

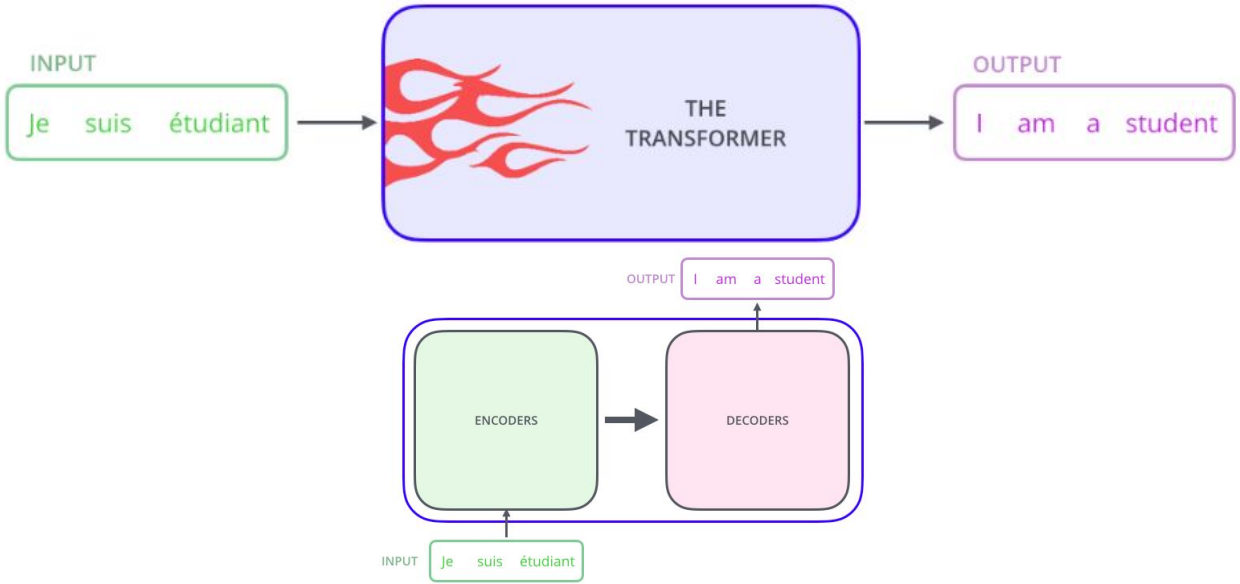
Attention Is All You Need- Vaswani et. al, Google 2017

- The **dominant** sequence transduction models are based on complex **recurrent or convolutional** neural networks that include an encoder and a decoder.
- The **best** performing models also connect the **encoder and decoder** through an **attention mechanism**.
- Propose a new simple network architecture, **“The Transformer”**, based **solely on attention mechanisms, dispensing with recurrence and convolutions entirely.**
- **superior** in quality while being **more parallelizable** and requiring **significantly less time to train.**



ASR Models - Transformer

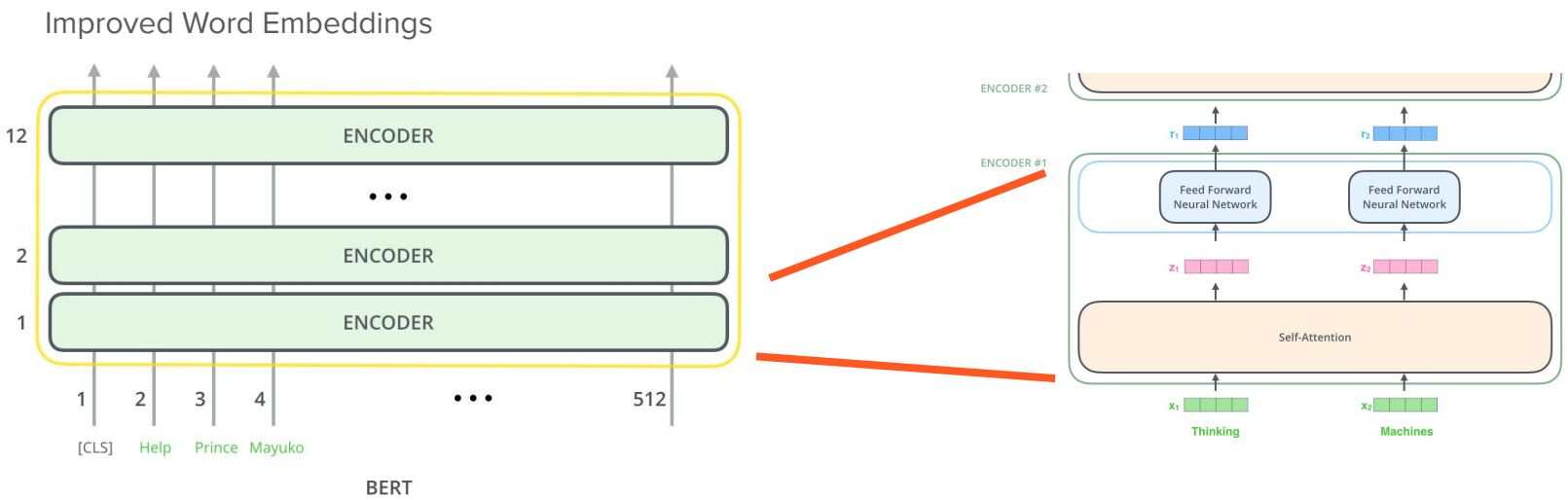
High Level Overview



We highly recommend reading: http://jalammar.github.io/illustrated-transformer/

ASR Models - Transformer

High Level Overview



We highly recommend reading: http://jalammar.github.io/illustrated-bert/

ASR Models - Transformer

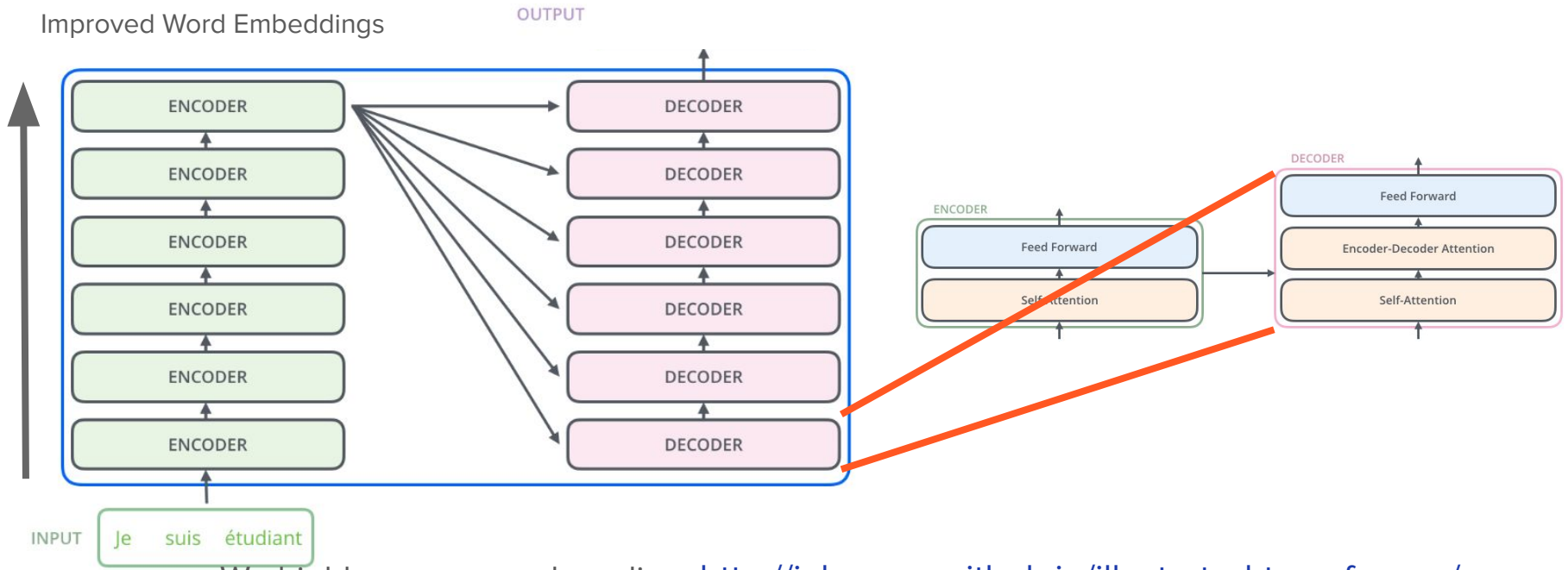
High Level Overview

We highly recommend reading: http://jalammar.github.io/illustrated-transformer/

ASR Models - Transformer

High Level Overview

Improved Word Embeddings

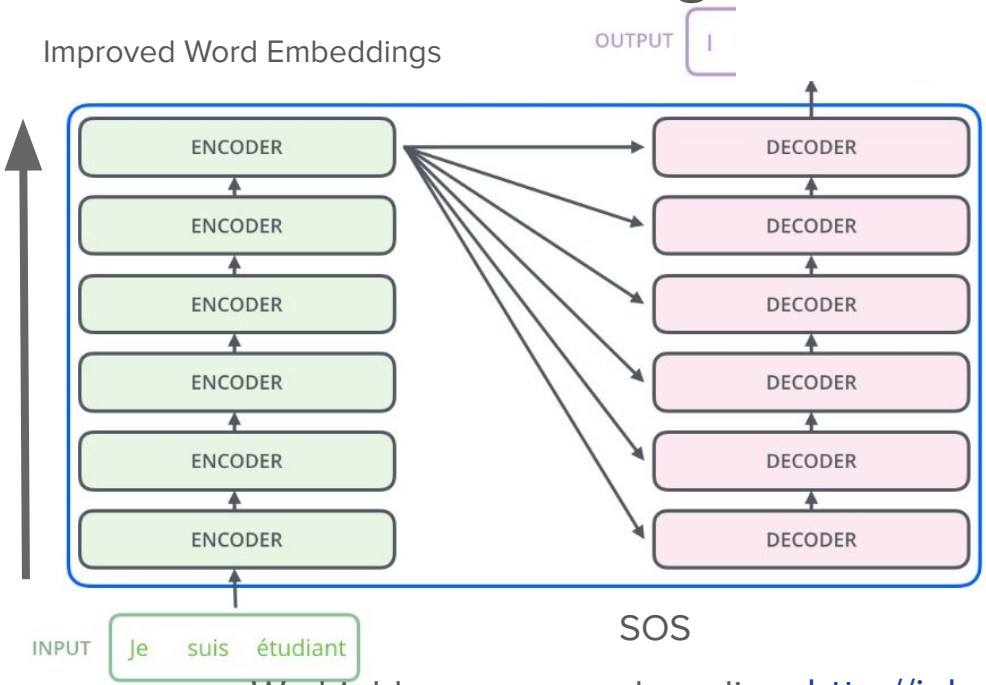


We highly recommend reading: http://jalammar.github.io/illustrated-transformer/

ASR Models - Transformer

High Level Overview

Improved Word Embeddings

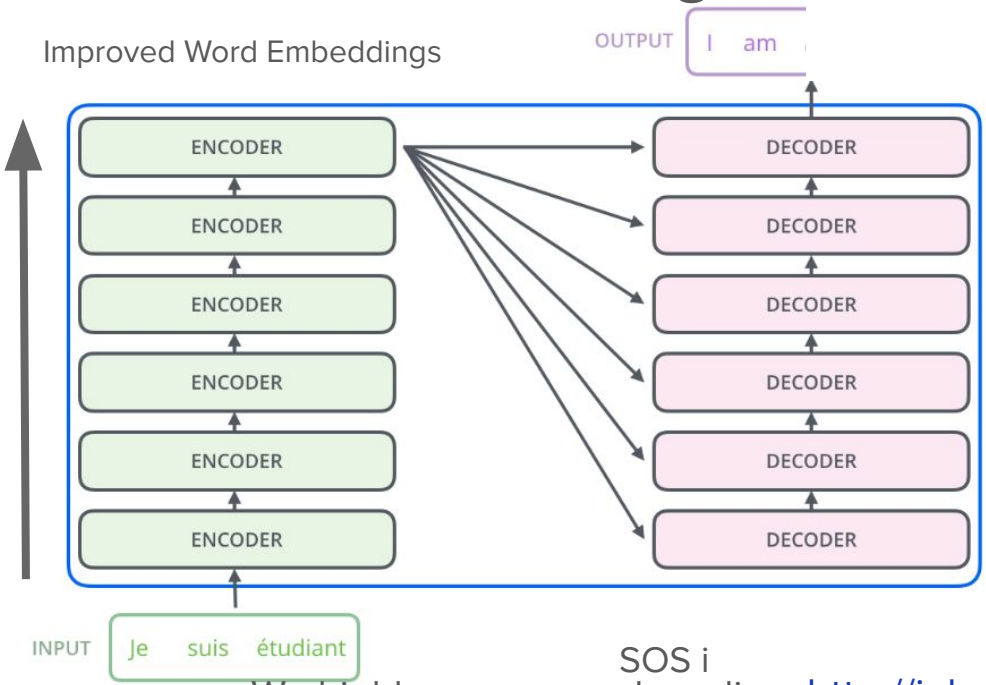


We highly recommend reading: http://jalammar.github.io/illustrated-transformer/

ASR Models - Transformer

High Level Overview

Improved Word Embeddings

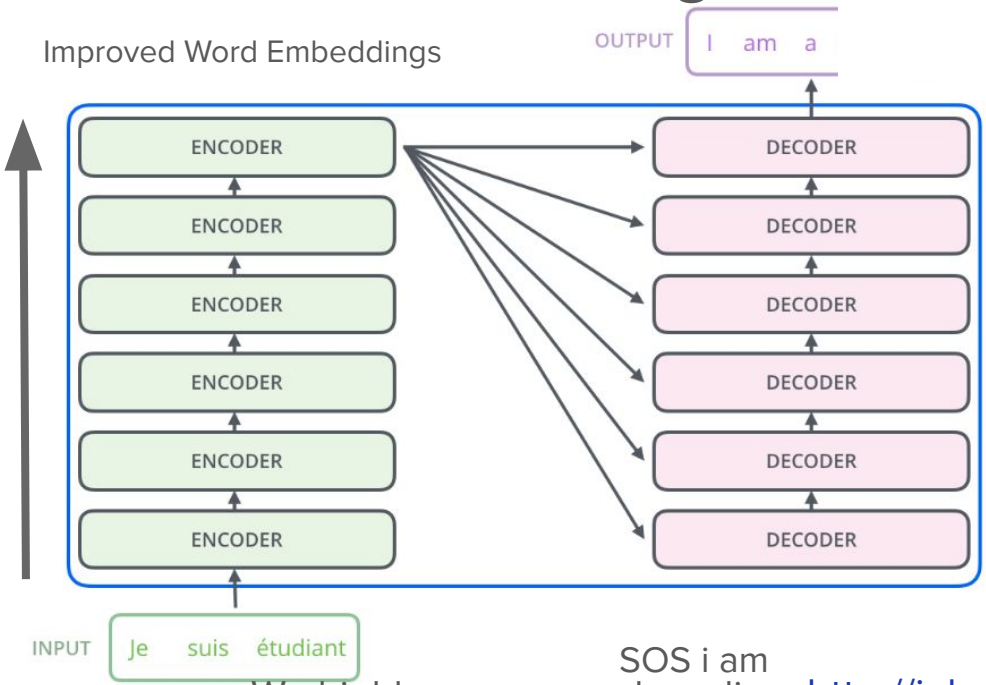


We highly recommend reading: http://jalammar.github.io/illustrated-transformer/

ASR Models - Transformer

High Level Overview

Improved Word Embeddings



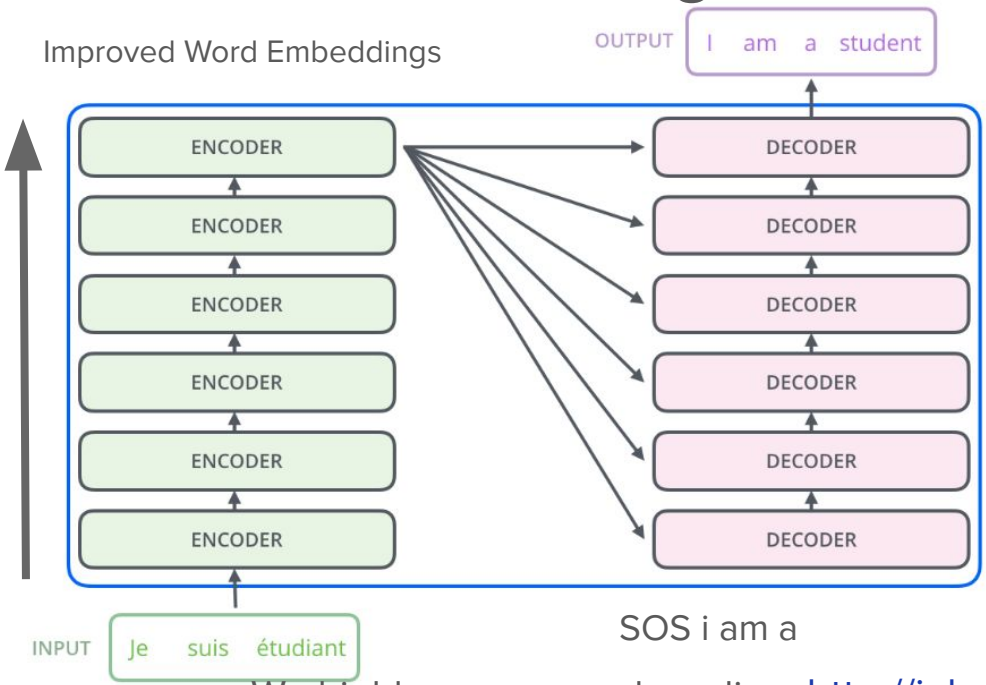
SOS i am

We highly recommend reading: http://jalammar.github.io/illustrated-transformer/

ASR Models - Transformer

High Level Overview

Improved Word Embeddings

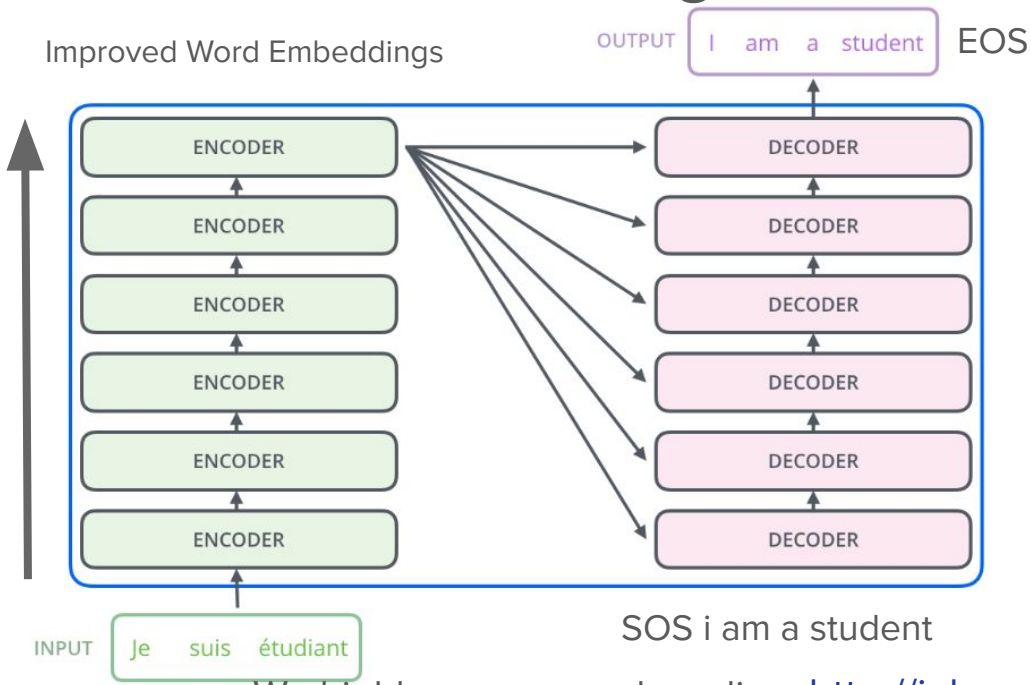


We highly recommend reading: http://jalammar.github.io/illustrated-transformer/

ASR Models - Transformer

High Level Overview

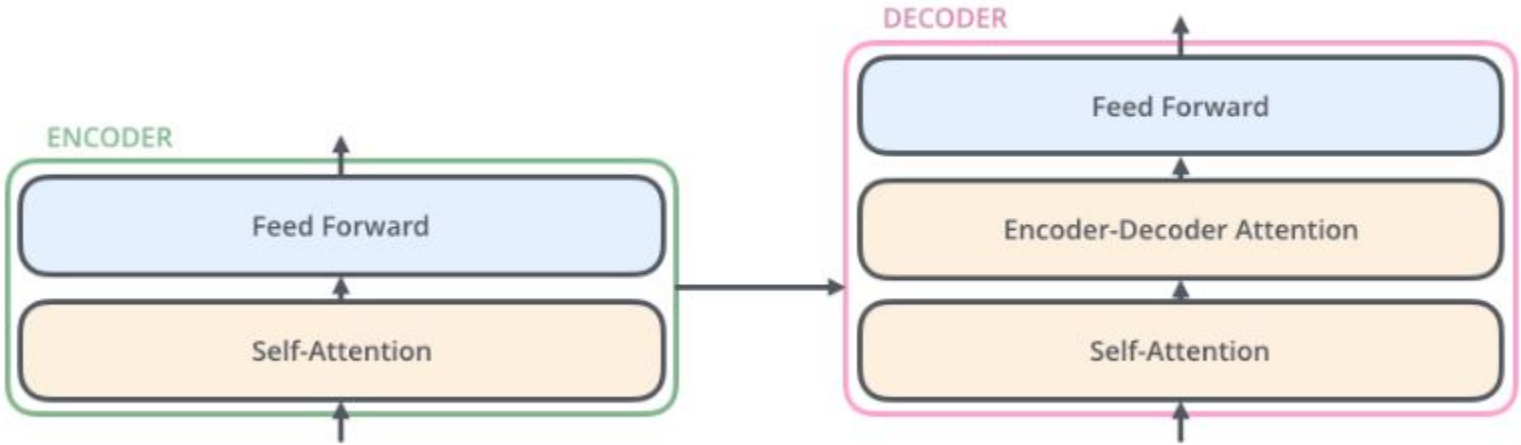
Improved Word Embeddings



We highly recommend reading: http://jalammar.github.io/illustrated-transformer/

ASR Models - Transformer

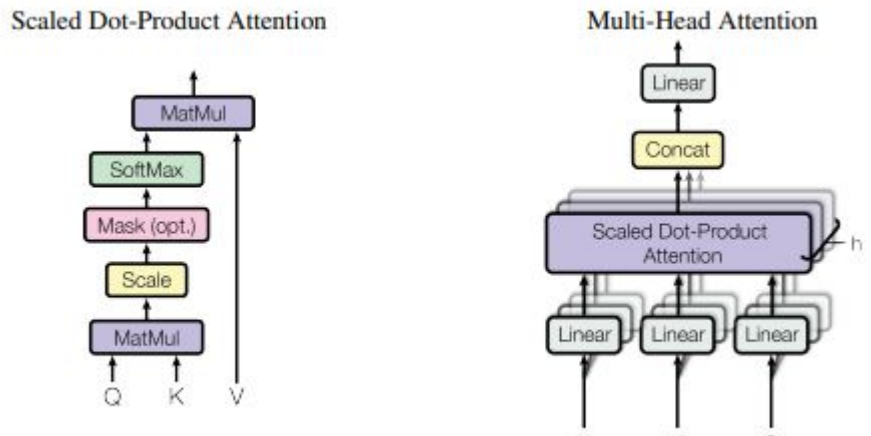
High Level Overview



We highly recommend reading: http://jalammar.github.io/illustrated-transformer/

ASR Models - Transformer

Attention- serves as a skip connection



$$\text{Attention}(Q, K, V) = \text{softmax}(\frac{QK^T}{\sqrt{d_k}})V$$

We highly recommend reading: http://jalammar.github.io/illustrated-transformer/

ASR Models - Transformer

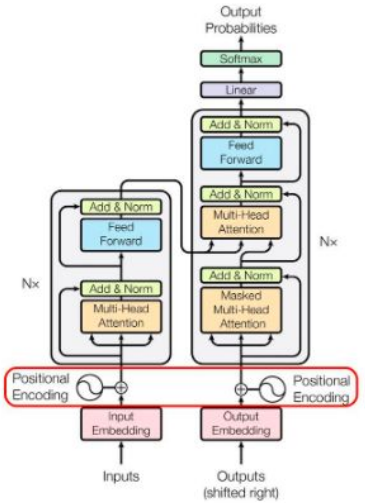
Positional Encodings

The intuition

You may wonder how this combination of sines and cosines could ever represent a position/order? It is actually quite simple, Suppose you want to represent a number in binary format, how will that be?

0 :	0	0	0	0	
1 :	0	0	0	1	
2 :	0	0	1	0	
3 :	0	0	1	1	
4 :	0	1	0	0	
5 :	0	1	0	1	
6 :	0	1	1	0	
7 :	0	1	1	1	
8 :	1	0	0	0	
9 :	1	0	0	1	
2 :	1	0	1	0	
11 :	1	0	1	1	
12 :	1	1	0	0	
13 :	1	1	0	1	
14 :	1	1	1	0	
15 :	1	1	1	1	

You can spot the rate of change between different bits. The LSB bit is alternating on every number, the second-lowest bit is rotating on every two numbers, and so on.



We highly recommend reading: http://jalammar.github.io/illustrated-transformer/

ASR Models - Transformer

Positional Encodings

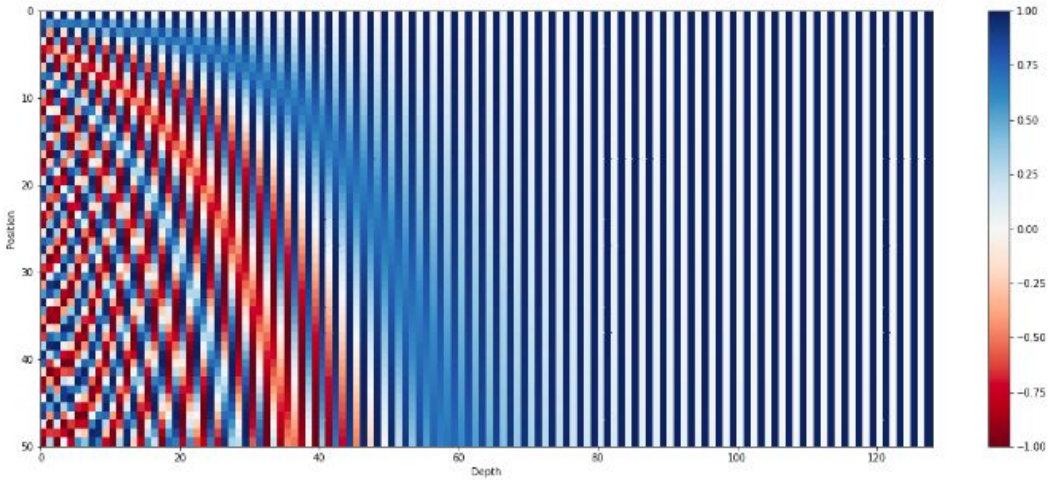


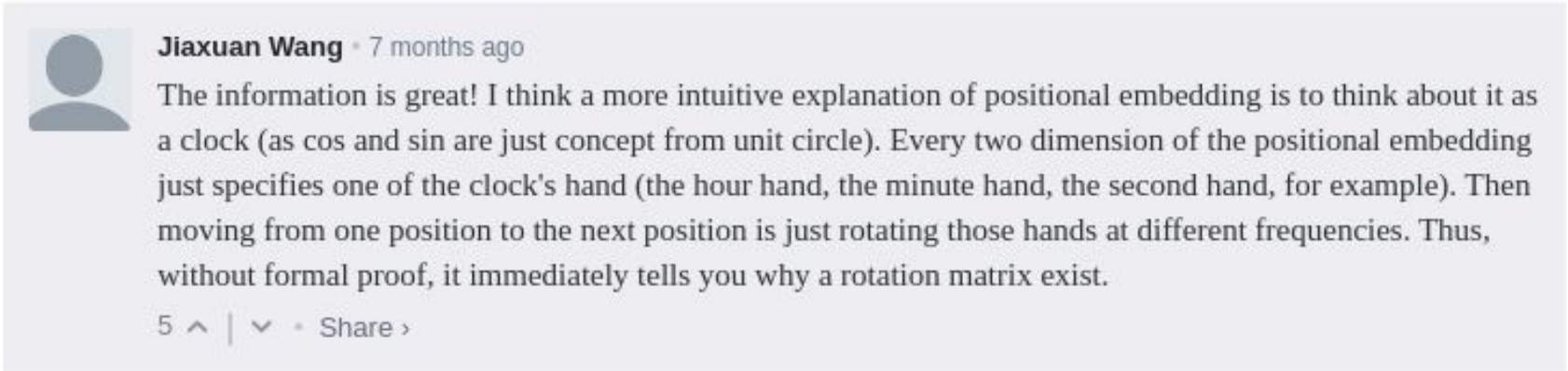
Figure 2 – The 128-dimensional positonal encoding for a sentence with the maximum lenght of 50. Each row represents the embedding vector $\overrightarrow{p_t}$

$$PE_{(pos,2i)} = sin(pos/10000^{2i/d_{model}})$$
$$PE_{(pos,2i+1)} = cos(pos/10000^{2i/d_{model}})$$

We highly recommend reading: http://jalammar.github.io/illustrated-transformer/

ASR Models - Transformer

Positional Encodings



Jiaxuan Wang • 7 months ago

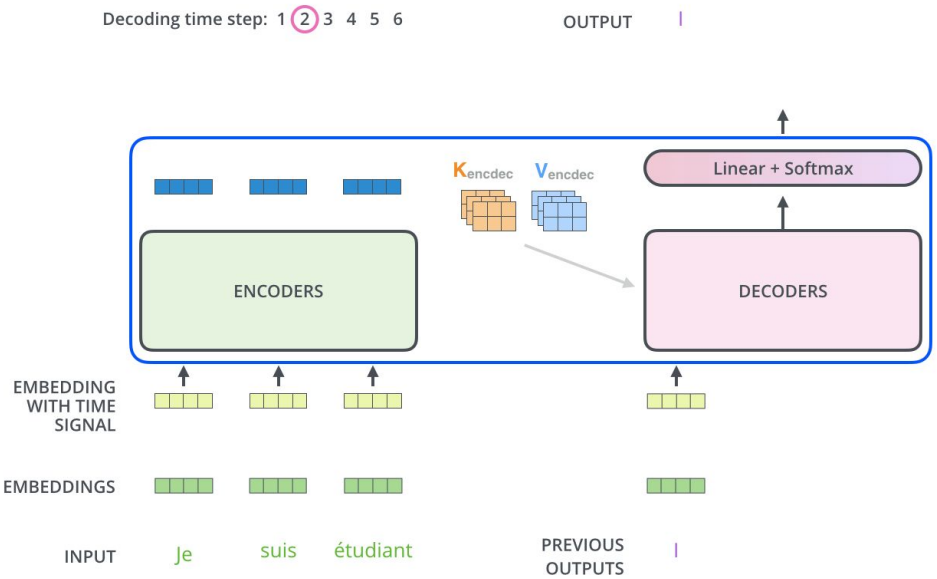
The information is great! I think a more intuitive explanation of positional embedding is to think about it as a clock (as cos and sin are just concept from unit circle). Every two dimension of the positional embedding just specifies one of the clock's hand (the hour hand, the minute hand, the second hand, for example). Then moving from one position to the next position is just rotating those hands at different frequencies. Thus, without formal proof, it immediately tells you why a rotation matrix exist.

5 ^ | v • Share ›

We highly recommend reading: https://kazemnejad.com/blog/transformer_architecture_positional_encoding/

ASR Models - Transformer

Cross Entropy Loss



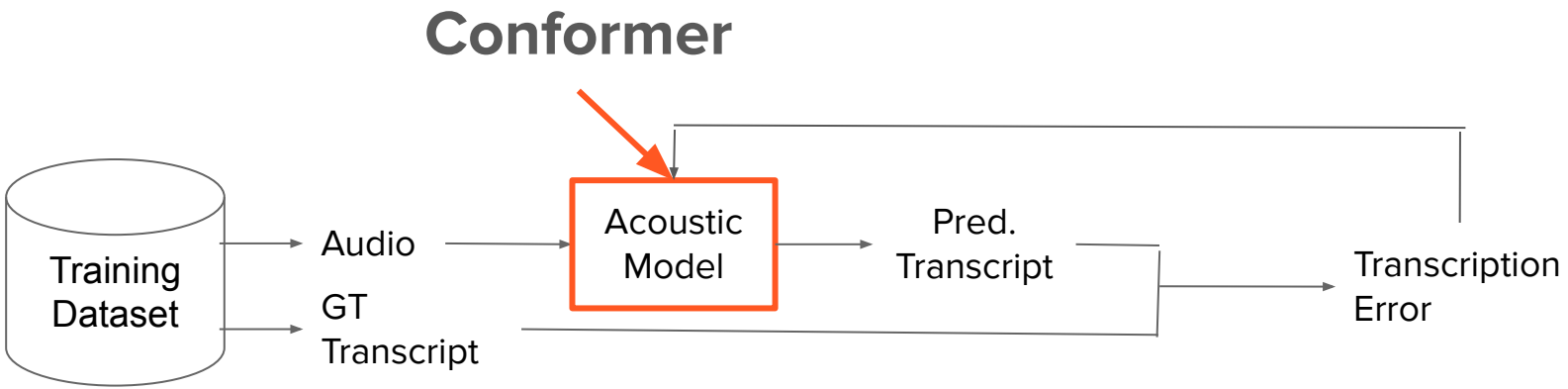
We highly recommend reading: http://jalammar.github.io/illustrated-transformer/

ASR Models - Transformer

Conformer: Convolution-augmented Transformer for Speech Recognition, Gulati, et. al, 2020, Google

Source 1

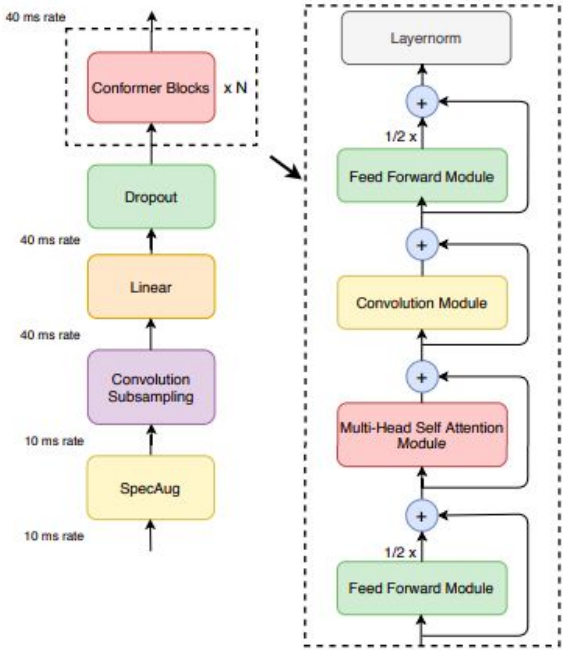
ASR Models - Transformer



ASR Models - Transformer

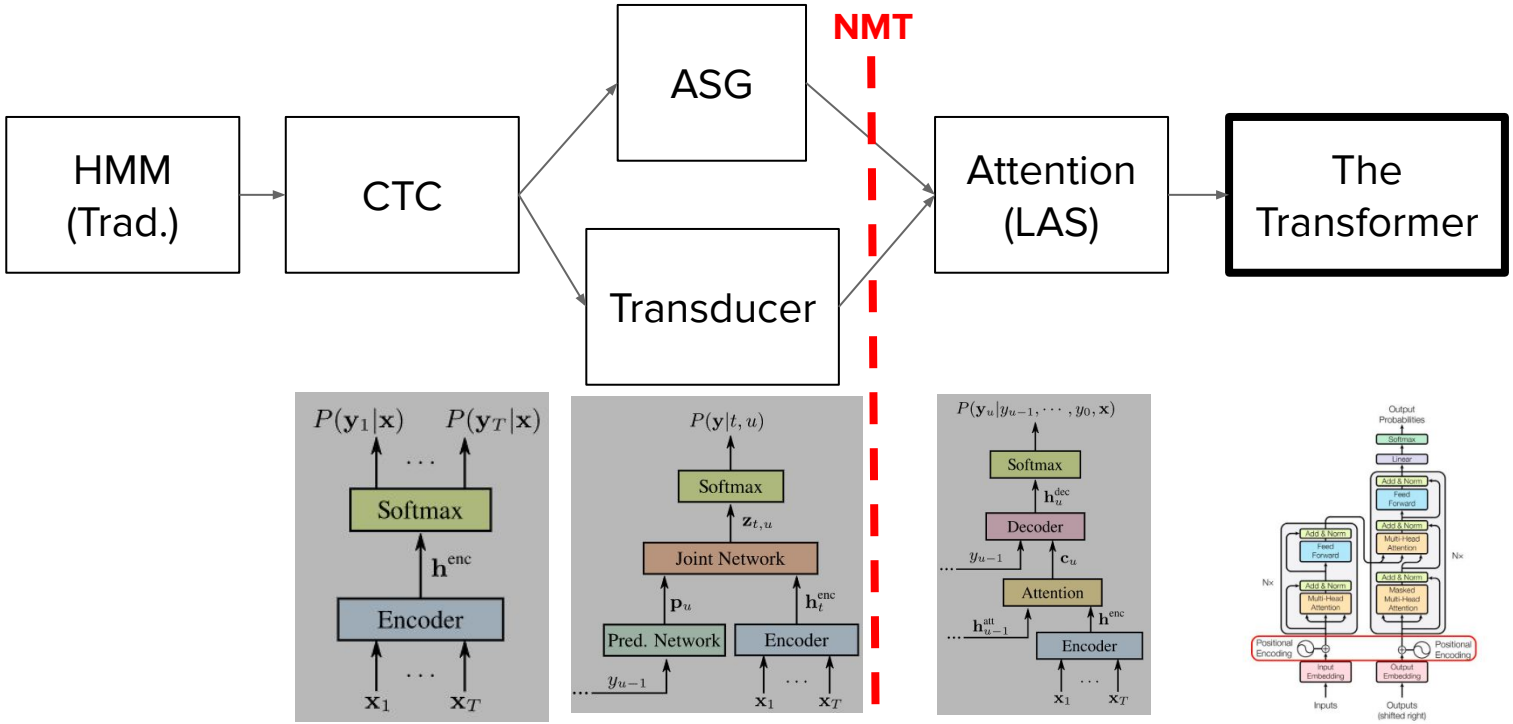
Conformer: Convolution-augmented Transformer for Speech Recognition, Gulati, et. al, 2020, Google

- **Transformer** models are good at capturing content-based **global interactions**
- **CNNs** exploit **local features** effectively.
- **Conformer** achieves the **best of both worlds** using a **combination** of self-attention and convolution, **sandwiched** between a pair feed forward modules.
- **Conformer** usually refers to the **encoder** (We use a single-LSTM-layer decoder in all our models)
- Streaming models are done using chunking



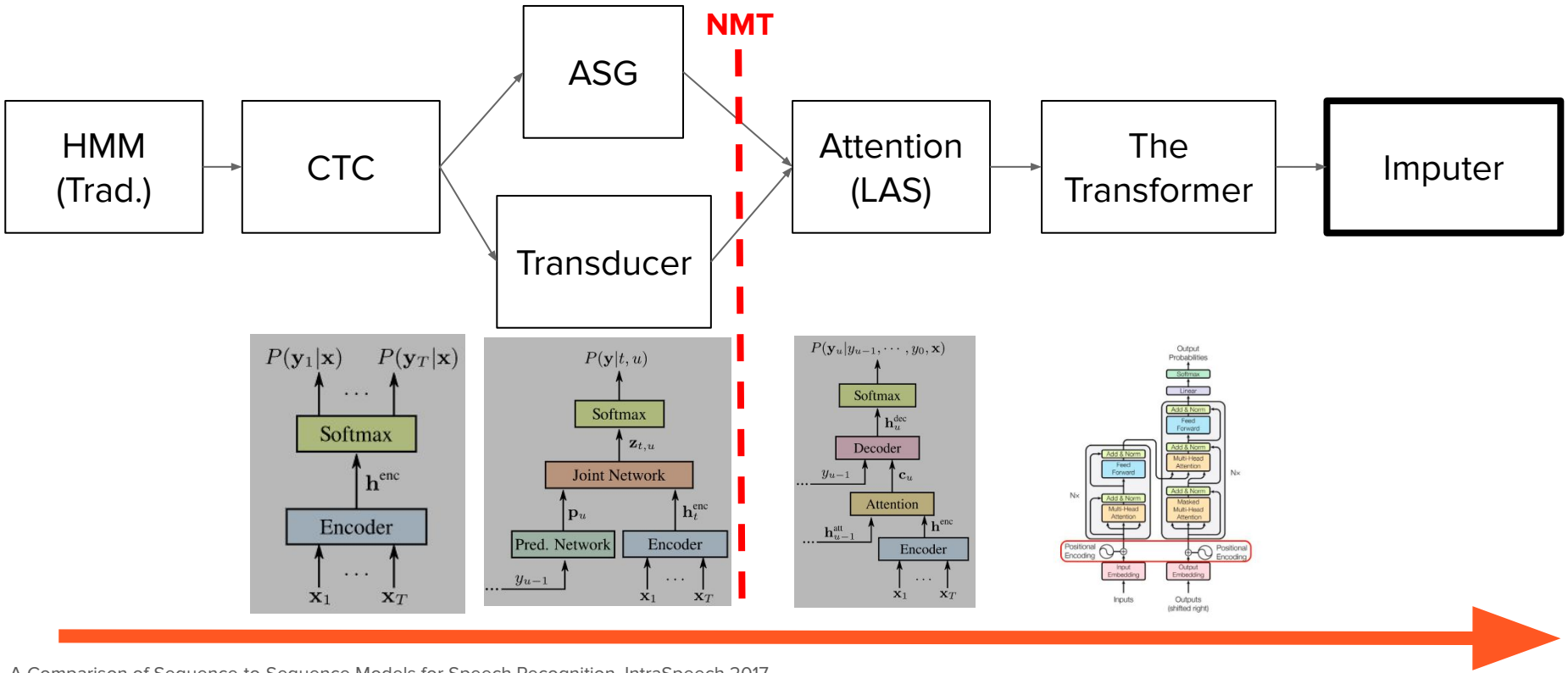
https://arxiv.org/pdf/2005.08100.pdf

ASR Models - Transformer



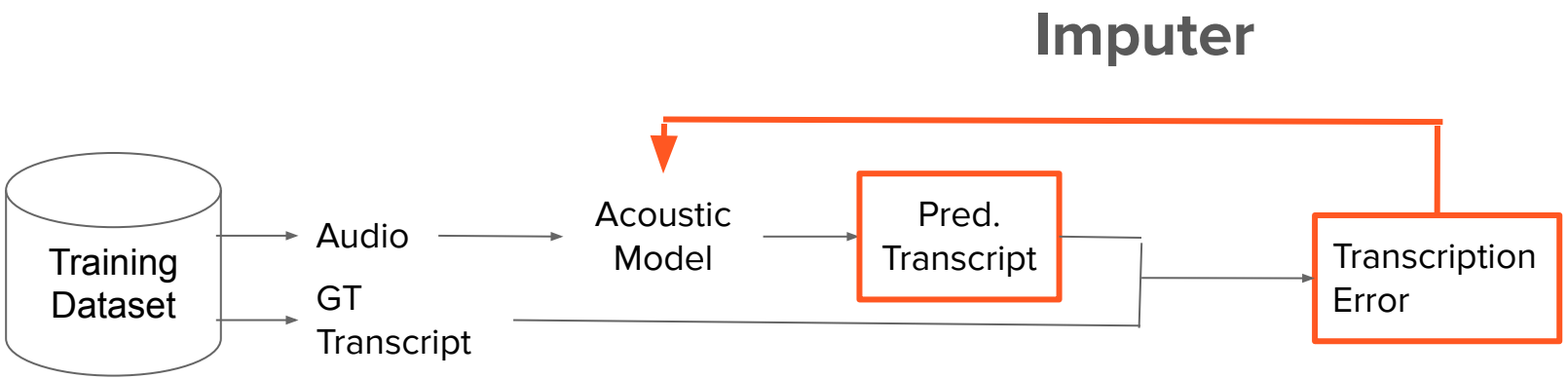
A Comparison of Sequence-to-Sequence Models for Speech Recognition. IntraSpeech 2017
Exploring Neural Transducers for End-to-End Speech Recognition (DeepSpeech 3) 2017
http://jalammar.github.io/illustrated-transformer/

ASR Models - Imputer



A Comparison of Sequence-to-Sequence Models for Speech Recognition. IntraSpeech 2017
Exploring Neural Transducers for End-to-End Speech Recognition (DeepSpeech 3) 2017
http://jalammar.github.io/illustrated-transformer/

ASR Models - Imputer



ASR Models - Imputer

Imputer: Sequence Modelling via Imputation and Dynamic Programming, Chan et. al, 2020

- Will not be covered, didn't get much attention.
- Similar to XLNet in principle- An iterative generative model, requiring only a **constant number of generation steps** independent of the number of input or output tokens. The Imputer can be trained to approximately marginalize over all possible alignments between the input and output sequences, **and all possible generation orders**

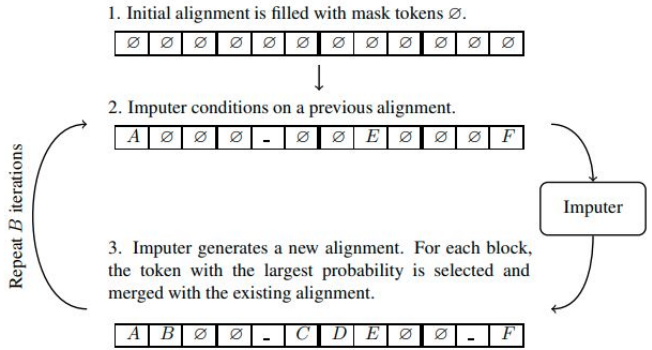
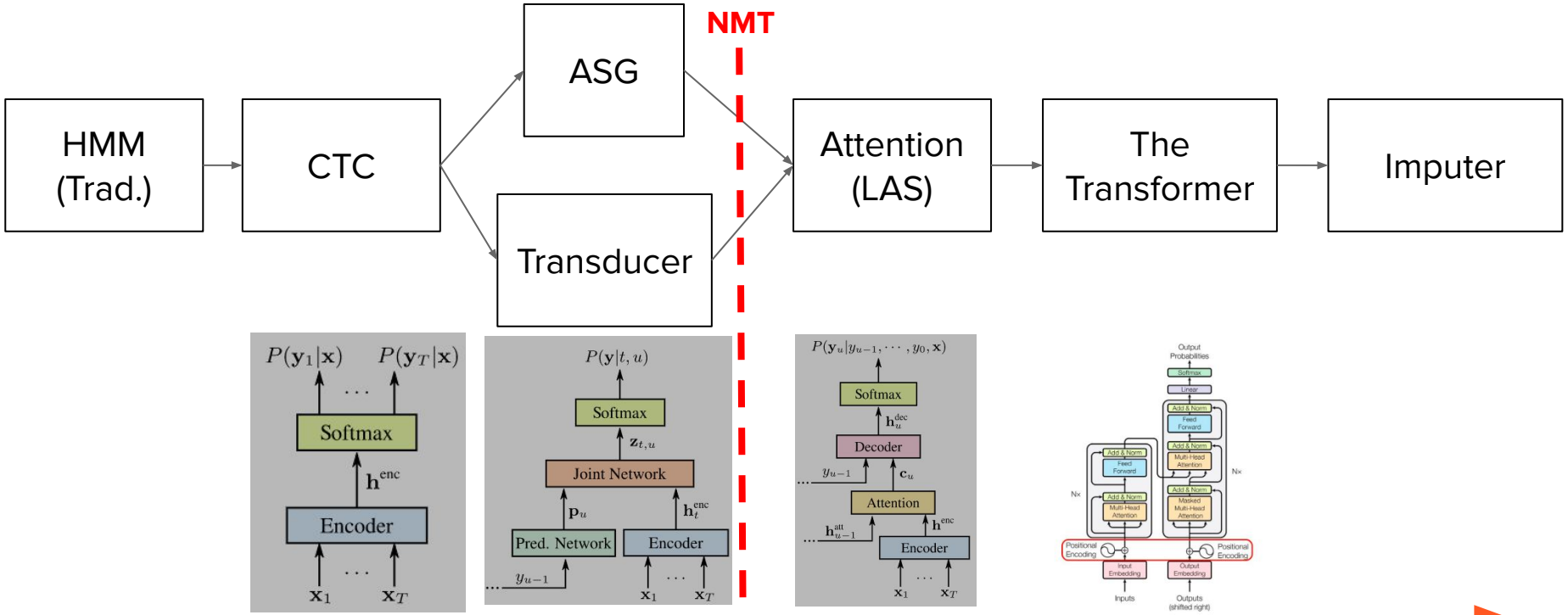


Figure 1. Visualization of the Imputer's decoding procedure. For this example, the alignment comprises 4 blocks with block size of $B = 3$ tokens each. The alignment "A B _ _ _ C D E _ _ _ F" is being imputed, with 1 token per block imputed at each decoding iteration. The decoding process takes exactly B iterations.

https://arxiv.org/pdf/2002.08926.pdf

ASR Models - Timeline



A Comparison of Sequence-to-Sequence Models for Speech Recognition. IntraSpeech 2017
Exploring Neural Transducers for End-to-End Speech Recognition (DeepSpeech 3) 2017
http://jalammar.github.io/illustrated-transformer/

Outline

- E2E framework
- Timeline
- **Benchmarks**
- Current Trends

ASR Models - Benchmarks

But how can we assess the robustness of a speech recognition system across diverse populations — young / old; men / women / nonbinary people; different ethnicities and cultural groups; and others?

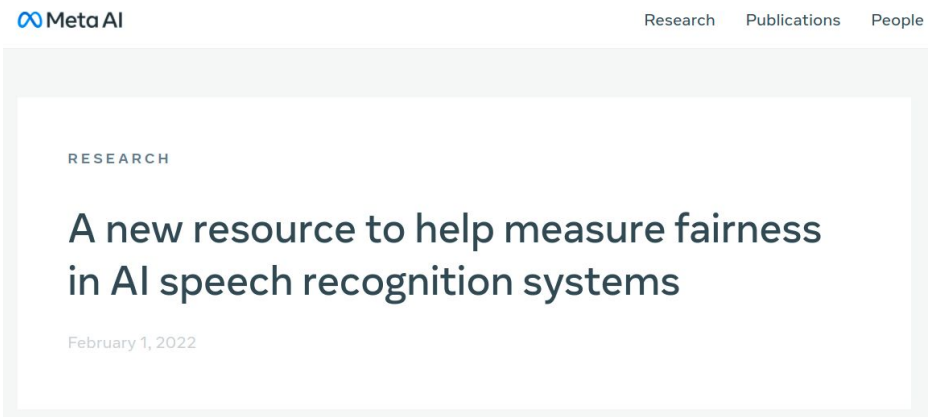


Fig. 2: **Example face crops** from the *Casual Conversations* dataset, categorized by their apparent Fitzpatrick skin types.

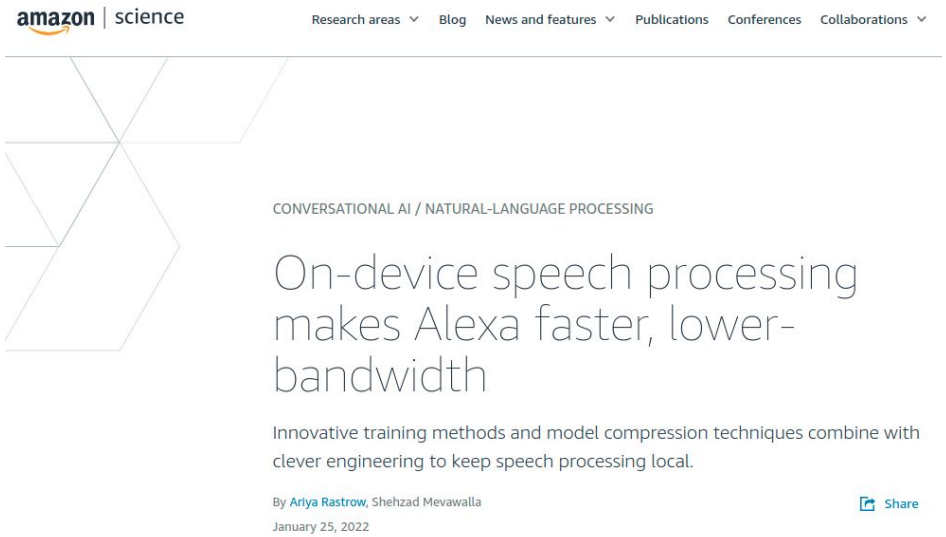
Source 1, 2

Outline

- E2E framework
- Timeline
- Benchmarks
- **Current Trends**
  - Embedded
  - Low latency
  - SSL (W2V, Hubert)
  - Whisper
  - ILM
  - Large scale
  - Unsupervised
  - Multi modal

Current Trends: Embedded (Streaming) ASR

"... [On device speech processing] has multiple benefits: a reduction in latency, or the time it takes Alexa to respond to queries; lowered bandwidth consumption, which is important on portable devices; and increased availability in in-car units and other applications where Internet connectivity is intermittent."



Source 1, 2

Current Trends: Low Latency ASR

FastEmit: LOW-LATENCY STREAMING ASR WITH SEQUENCE-LEVEL EMISSION REGULARIZATION- Google 2021

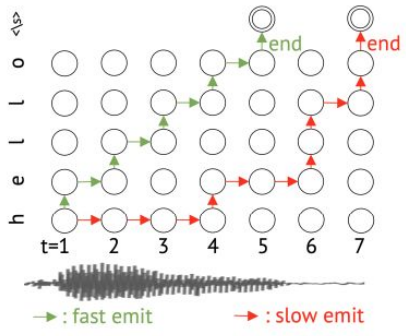


Fig. 1. Examples of fast and slow transducer emission lattices (speech-text alignments). Transducer aims to maximize the log-probability of any lattice, regardless of its emission latency.

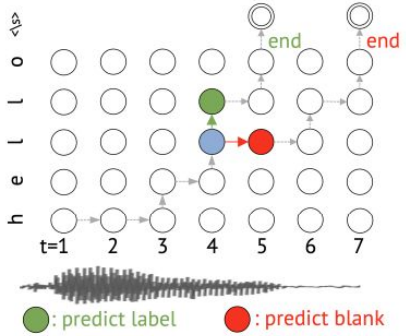
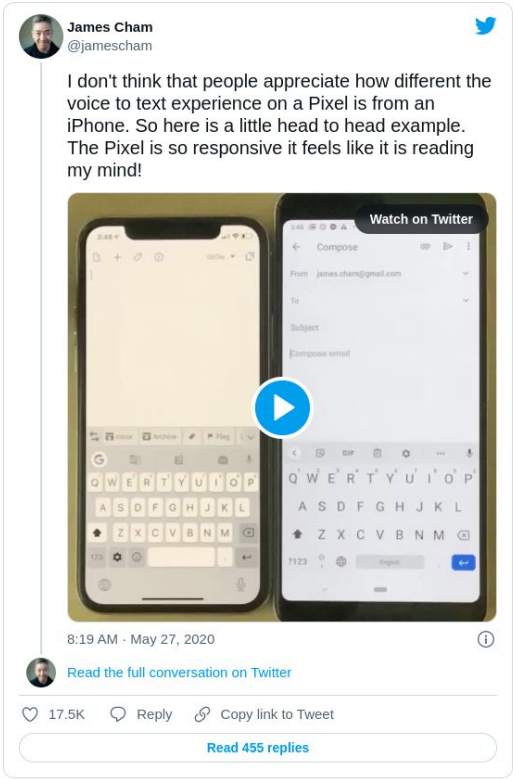


Fig. 2. Illustration of *FastEmit* regularization. Consider any node (*e.g.*, **blue** node), *FastEmit* encourages predicting label $y \in \mathcal{Y}$ (**green** node) instead of predicting blank $\varnothing$ (**red** node).

https://arxiv.org/pdf/2010.11148.pdf

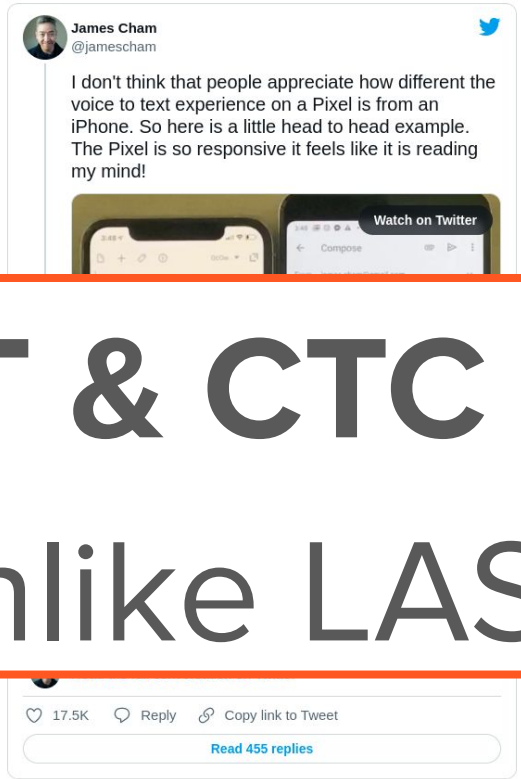


https://twitter.com/i/status/1265512829806927873

Current Trends: Low Latency ASR

FastEmit: LOW-LATENCY STREAMING ASR WITH SEQUENCE-LEVEL EMISSION REGULARIZATION- Google 2021

https://arxiv.org/pdf/2010.11148.pdf



https://twitter.com/i/status/1265512829806927873

That's why **RNN-T & CTC** are so common, unlike LAS

Current Trends: Low Latency ASR

Scaling Up Online Speech Recognition Using ConvNets

Vineel Pratap, Qiantong Xu, Jacob Kahn, Gilad Avidov, Tatiana Likhomanenko, Awni Hannun, Vitaliy Liptchinsky, Gabriel Synnaeve, Ronan Collobert

Facebook AI Research

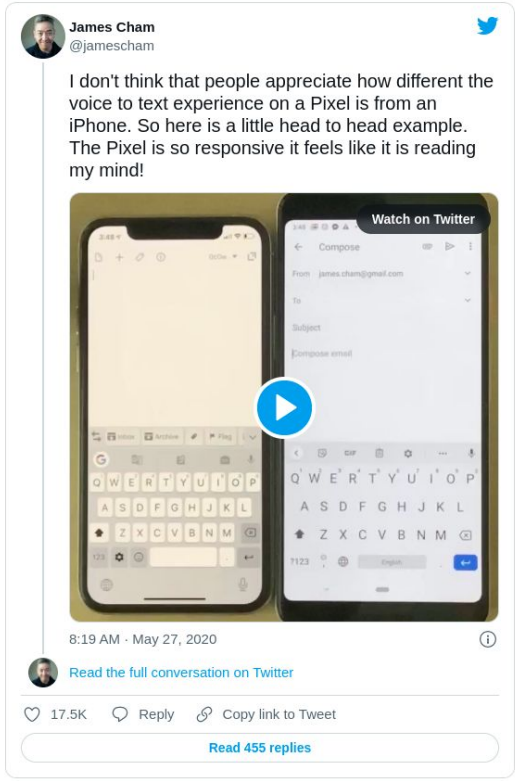
PEAK-FIRST CTC: REDUCING THE PEAK LATENCY OF CTC MODELS BY APPLYING PEAK-FIRST REGULARIZATION

Zhengkun Tian, Hongyu Xiang, Min Li, Feifei Lin, Ke Ding, Guanglu Wan

Meituan, Beijing, China

$$\mathcal{L}_{PFR} = \sum_{t=2}^{T} \boldsymbol{p}^{t+1} \log \frac{\boldsymbol{p}^{t+1}}{\boldsymbol{p}^{t}} = \sum_{t=2}^{T} \sum_{k=1}^{V} p_k^{t+1} \log \frac{p_k^{t+1}}{p_k^{t}} \quad (3)$$

where $\boldsymbol{p}^t$ is the output probability distribution over the vocabulary, and V is the size of vocabulary. In the distillation process, the too



https://twitter.com/i/status/1265512829806927873

 https://arxiv.org/pdf/2001.09727.pdf https://arxiv.org/pdf/2211.03284.pdf

Current Trends: SSL

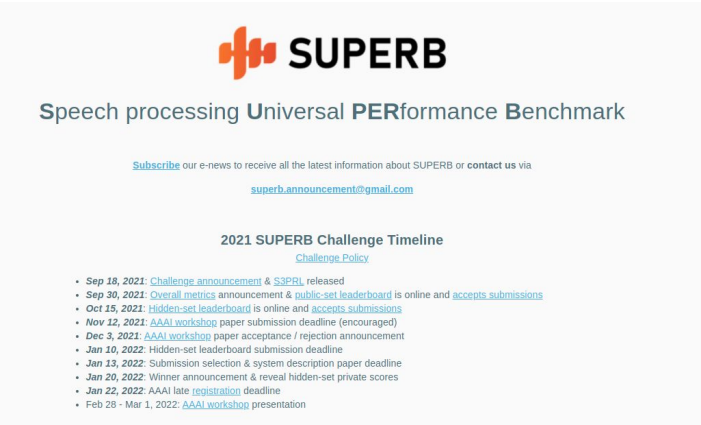
semi supervised learning (SSL) means pre training on unsupervised tasked followed by fine tuning on supervised data.

Self training is a simple semi-supervised learning approach: Unlabelled examples that attract high-confidence predictions are labelled with their predictions and added to the training set

Current Trends: SSL- Audio Embeddings

“Self-supervised learning (SSL) has proven vital for advancing research in natural language processing (NLP) and computer vision (CV). The paradigm pretrains a shared model on large volumes of unlabeled data and achieves state-of-the-art (SOTA) for various tasks with minimal adaptation.

... We introduce Speech processing Universal PERformance Benchmark (SUPERB). SUPERB is a leaderboard to benchmark the performance of a shared model across a wide range of speech processing tasks with minimal architecture changes and labeled data. ”



https://superbbenchmark.org/

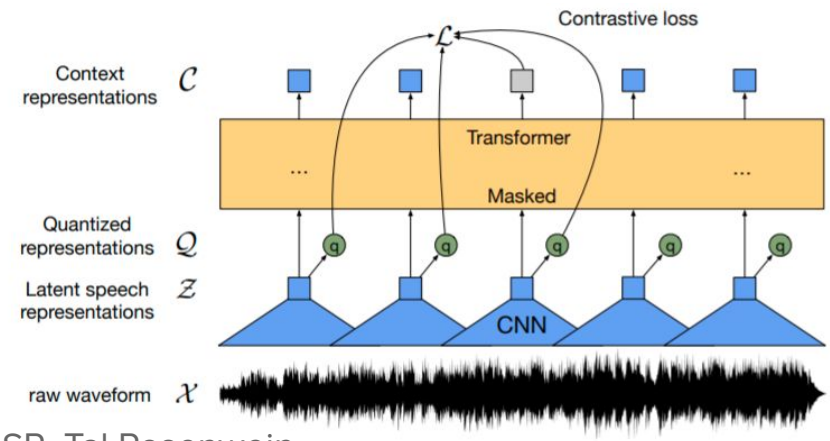
Current Trends: SSL- Pre training

wav2vec 2.0: A Framework for Self-Supervised Learning of Speech Representations

Alexei Baevski Henry Zhou Abdelrahman Mohamed Michael Auli

{abaevski,henryzhou7,abdo,michaelauli}@fb.com

Facebook AI



HuBERT: Self-Supervised Speech Representation Learning by Masked Prediction of Hidden Units

Wei-Ning Hsu, Benjamin Bolte, Yao-Hung Hubert Tsai, Kushal Lakhotia, Ruslan Salakhutdinov, Abdelrahman Mohamed

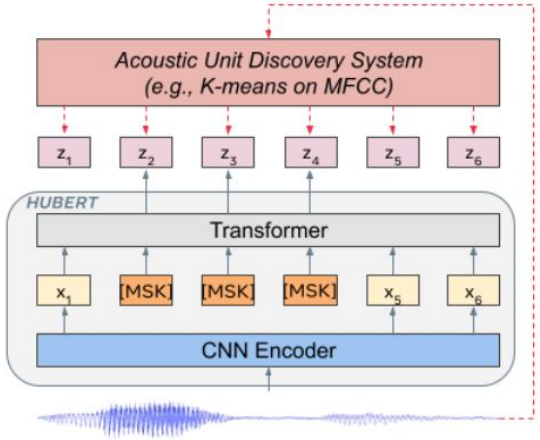


Fig. 1: The HuBERT approach predicts hidden cluster assignments of the masked frames (y_2, y_3, y_4 in the figure) generated by one or more iterations of k-means clustering.

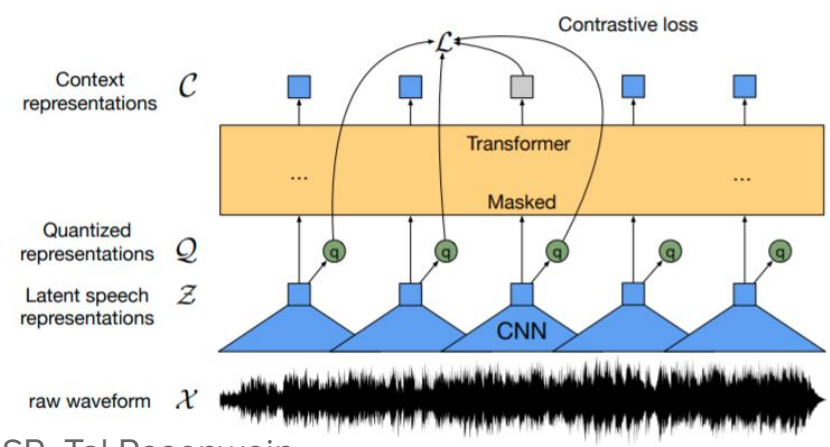
Current Trends: SSL- Wav2Vec2- Why?

wav2vec 2.0: A Framework for Self-Supervised Learning of Speech Representations

Alexei Baevski **Henry Zhou** **Abdelrahman Mohamed** **Michael Auli**

{abaevski,henryzhou7,abdo,michaelauli}@fb.com

Facebook AI



- current speech recognition systems **require thousands of hours** of transcribed speech to reach acceptable performance which is **not available for** the vast majority of the nearly 7,000 **languages spoken worldwide**.
- Learning purely from labeled examples **does not resemble language acquisition in humans**: infants learn language by listening to adults around them - a process that requires learning good representations of speech.

Current Trends: SSL- Wav2Vec2- What?

We show for the first time that **learning powerful representations** from speech audio alone **followed by fine-tuning** on transcribed speech can **outperform** the best **semi-supervised methods** while being **conceptually simpler.**

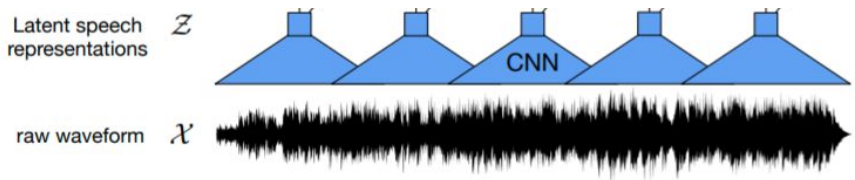
- Experiments using **all labeled data** of Librispeech achieve **1.8/3.3 WER** on the clean/other test sets.
- When lowering the amount of labeled data to **one hour**, **wav2vec 2.0 outperforms** the previous state of the art on **the 100 hour subset** while using 100 times less labeled data.
- Using just **ten minutes** of labeled data and **pre-training on 53k hours** of unlabeled data still achieves **4.8/8.2 WER**.

Current Trends: SSL- Wav2Vec2- How?

- **Pre-Training:** Our approach **encodes speech** audio **via** a multi-layer **convolutional neural network** and **then masks spans** of the resulting **latent speech representations** [26, 56], similar to masked language modeling [9]. The latent representations are **fed to a Transformer** network to build **contextualized representations** and the model is trained via a **contrastive task** where the true latent is to be distinguished from distractors
- **Fine Tuning:** After pre-training on unlabeled speech, the model is **fine-tuned on labeled data** with a Connectionist Temporal Classification **(CTC) loss** [14, 4] to be used for downstream speech recognition tasks

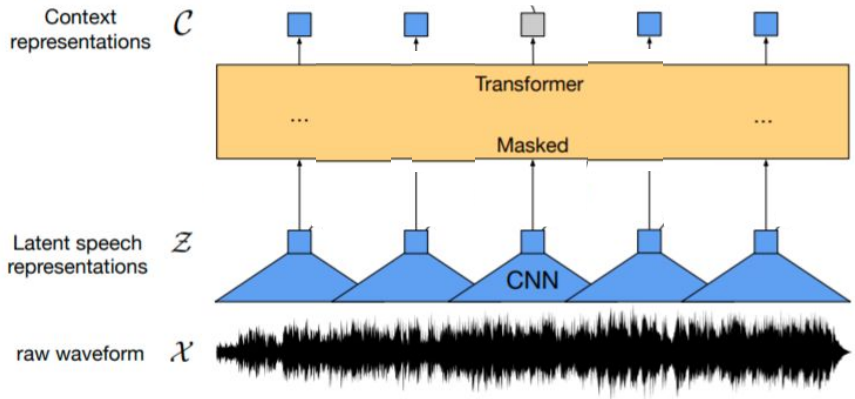
Current Trends: SSL- Wav2Vec2- Model

Our model is composed of a multi-layer convolutional feature encoder $f : \mathcal{X} \mapsto \mathcal{Z}$ which takes as input raw audio $\mathcal{X}$ and outputs latent speech representations $\mathbf{z}_1, \ldots, \mathbf{z}_T$ for T time-steps.



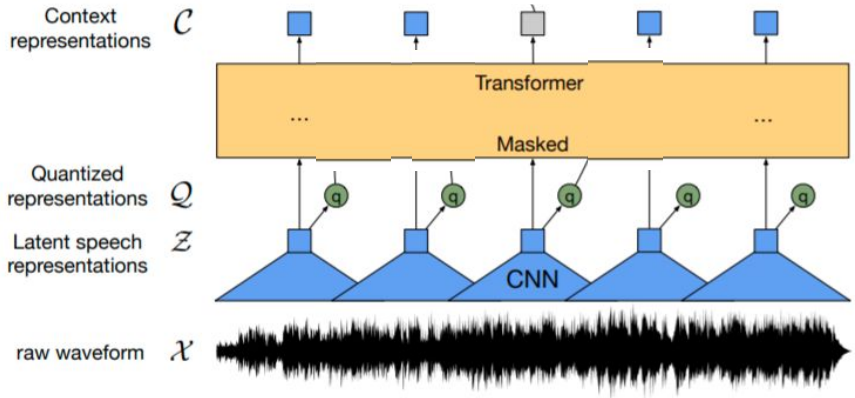
Current Trends: SSL- Wav2Vec2- Model

Our model is composed of a multi-layer convolutional feature encoder $f : \mathcal{X} \mapsto \mathcal{Z}$ which takes as input raw audio $\mathcal{X}$ and outputs latent speech representations $\mathbf{z}_1, \ldots, \mathbf{z}_T$ for T time-steps. They are then fed to a Transformer $g : \mathcal{Z} \mapsto \mathcal{C}$ to build representations $\mathbf{c}_1, \ldots, \mathbf{c}_T$ capturing information from the entire sequence [9, 5, 4].



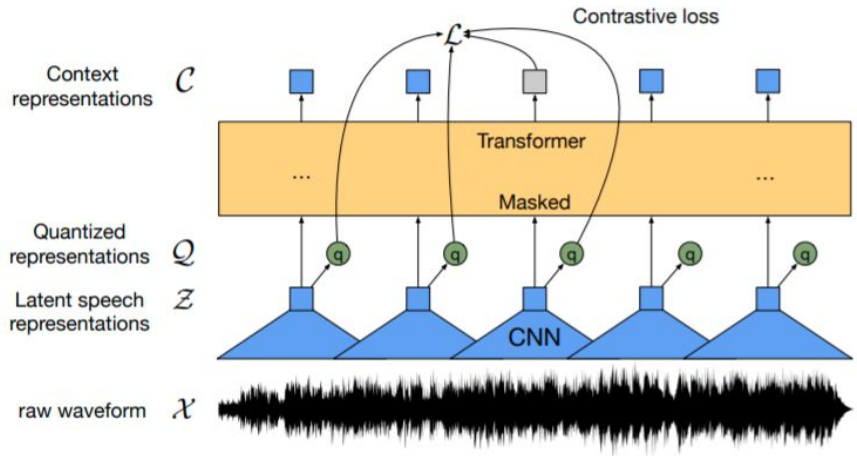
Current Trends: SSL- Wav2Vec2- Model

Our model is composed of a multi-layer convolutional feature encoder $f : \mathcal{X} \mapsto \mathcal{Z}$ which takes as input raw audio $\mathcal{X}$ and outputs latent speech representations $\mathbf{z}_1, \ldots, \mathbf{z}_T$ for T time-steps. They are then fed to a Transformer $g : \mathcal{Z} \mapsto \mathcal{C}$ to build representations $\mathbf{c}_1, \ldots, \mathbf{c}_T$ capturing information from the entire sequence [9, 5, 4]. The output of the feature encoder is discretized to $\mathbf{q}_t$ with a quantization module $\mathcal{Z} \mapsto \mathcal{Q}$



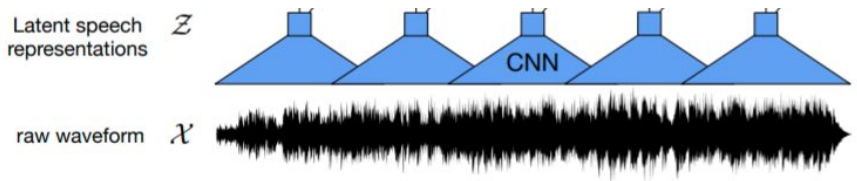
Current Trends: SSL- Wav2Vec2- Model

Our model is composed of a multi-layer convolutional feature encoder $f : \mathcal{X} \mapsto \mathcal{Z}$ which takes as input raw audio $\mathcal{X}$ and outputs latent speech representations $\mathbf{z}_1, \ldots, \mathbf{z}_T$ for T time-steps. They are then fed to a Transformer $g : \mathcal{Z} \mapsto \mathcal{C}$ to build representations $\mathbf{c}_1, \ldots, \mathbf{c}_T$ capturing information from the entire sequence [9, 5, 4]. The output of the feature encoder is discretized to $\mathbf{q}_t$ with a quantization module $\mathcal{Z} \mapsto \mathcal{Q}$ to represent the targets (Figure 1) in the self-supervised objective



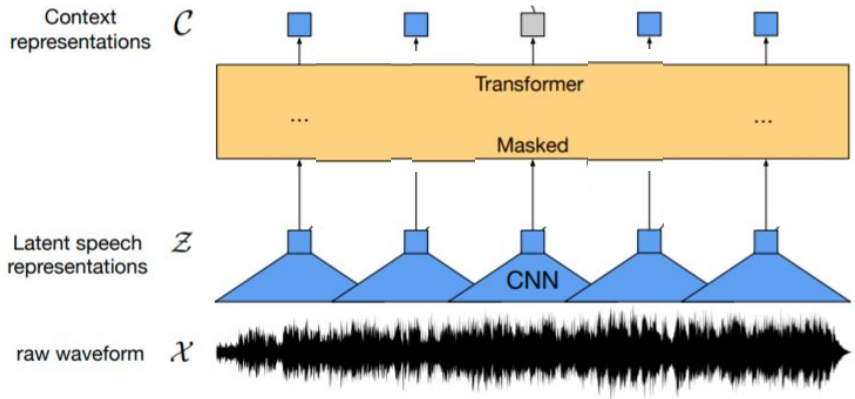
Current Trends: SSL- Wav2Vec2- Model- Feature Encoder

- The **encoder consists of** several blocks containing a **temporal convolution** followed by layer normalization [1] and a GELU activation function [21].
- The raw waveform **input** to the encoder is **normalized** to zero mean and unit variance.
- This results in an encoder output frequency of 49 hz with a **stride of about 20ms** between each sample, and a receptive field of 400 input samples or **25ms of audio.**



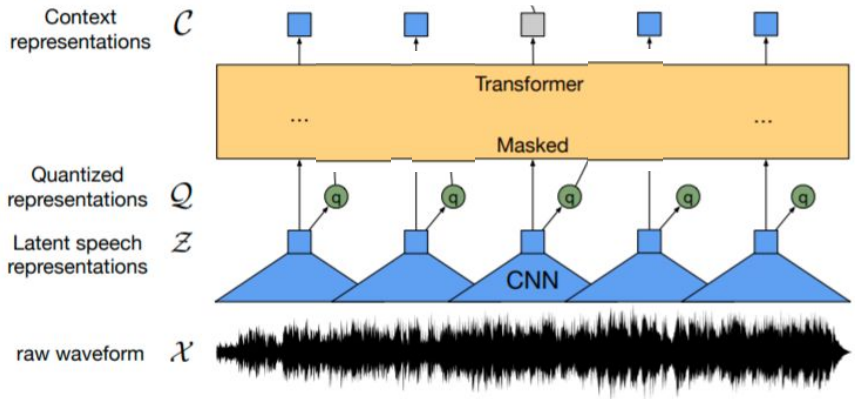
Current Trends: SSL- Wav2Vec2- Model-
Contextualized representations with Transformers.

- **The output of the feature encoder is fed to** a context network which follows the Transformer architecture- **'BERT style' encoder** [55, 9, 33].
- Use a **convolutional layer** (on the feature encoder outputs) which acts as **relative positional embedding**.



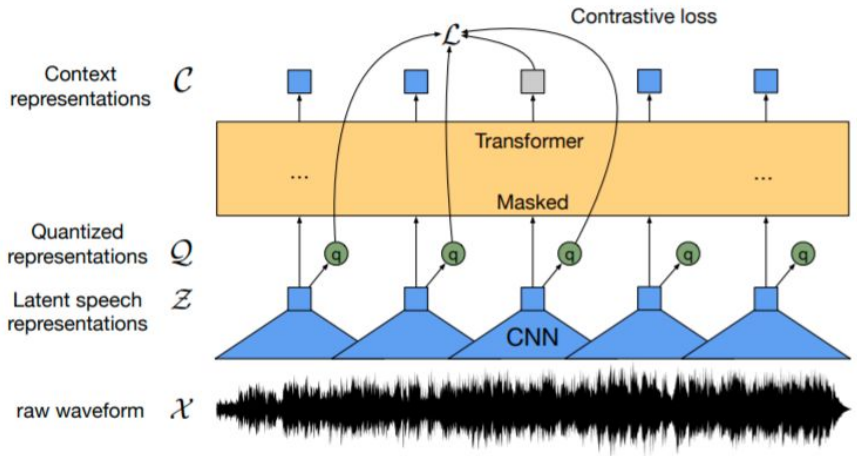
Current Trends: SSL- Wav2Vec2- Model- Quantization Module

- For self-supervised training we discretize the output of the feature encoder z to a finite set of speech representations via product quantization.



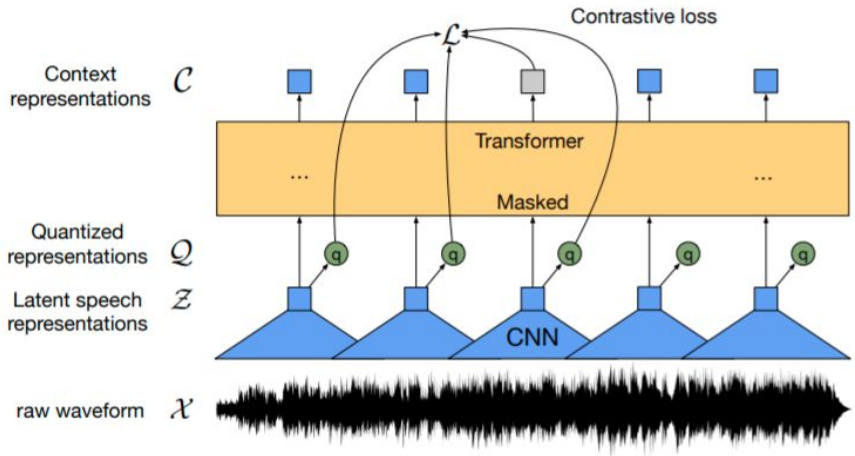
Current Trends: SSL- Wav2Vec2- Model- Training: Masking

- To pre-train the model we **mask** a certain proportion of **time steps** in the latent **(continuous) feature encoder space**, similar to masked language modeling in BERT.
- randomly sample without replacement *p*% of all time steps and then mask the subsequent *M* consecutive time steps from every sampled index.



Current Trends: SSL- Wav2Vec2- Model- Training: Objective

- **Contrastive Loss:** Given the **(continuous) output of the Transformer**, the loss requires **identifying** the **correct quantized** latent audio **representation** in a **set of distractors** for each **masked time step**
- **Codebook Diversity Loss:** maximizing the entropy over the codebook entries. encourages the model to **use the codebook entries equally often**.



Current Trends: SSL- Wav2Vec2- Model- Fine-Tuning

- Pre-trained models are **fine-tuned** for speech recognition **by adding a randomly initialized linear projection** on top of the **context network** into C classes representing the vocabulary of the task [4].
- Models are **optimized** by minimizing **a CTC loss**

Table 4: Average WER and standard deviation on combined dev-clean/other of Librispeech for three training seeds. We ablate quantizing the context network input and the targets in the contrastive loss.

	avg. WER	std.
Continuous inputs, quantized targets (Baseline)	7.97	0.02
Quantized inputs, quantized targets	12.18	0.41
Quantized inputs, continuous targets	11.18	0.16
Continuous inputs, continuous targets	8.58	0.08

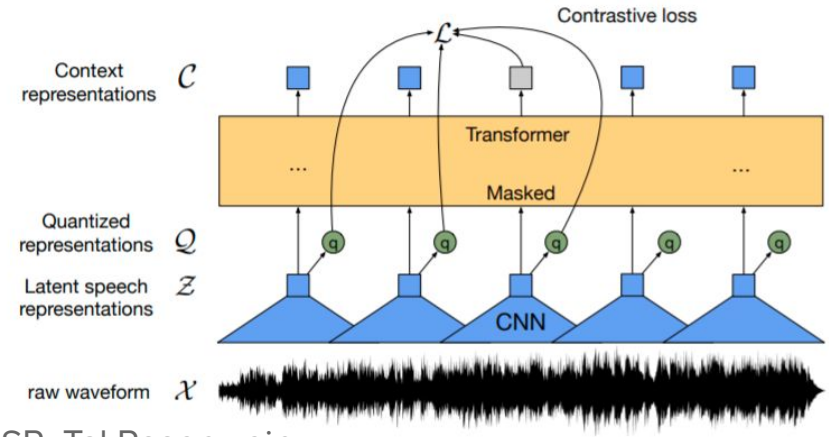
Current Trends: SSL- Pre training

wav2vec 2.0: A Framework for Self-Supervised Learning of Speech Representations

Alexei Baevski Henry Zhou Abdelrahman Mohamed Michael Auli

{abaevski,henryzhou7,abdo,michaelauli}@fb.com

Facebook AI



HuBERT: Self-Supervised Speech Representation Learning by Masked Prediction of Hidden Units

Wei-Ning Hsu, Benjamin Bolte, Yao-Hung Hubert Tsai, Kushal Lakhotia, Ruslan Salakhutdinov, Abdelrahman Mohamed

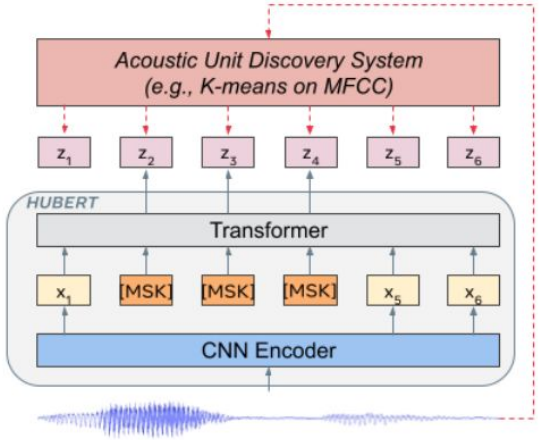


Fig. 1: The HuBERT approach predicts hidden cluster assignments of the masked frames (y_2, y_3, y_4 in the figure) generated by one or more iterations of k-means clustering.

Current Trends: SSL- HuBERT

We propose the Hidden-Unit BERT (HuBERT) approach for self-supervised speech representation learning, which utilizes **an offline clustering step** to provide **aligned target labels** for a **BERT-like prediction loss**. A key ingredient of our approach is applying the **prediction loss over the masked regions only**, which forces the model to learn a **combined acoustic and language model** over the continuous inputs.

HuBERT: Self-Supervised Speech Representation Learning by Masked Prediction of Hidden Units

Wei-Ning Hsu, Benjamin Bolte, Yao-Hung Hubert Tsai, Kushal Lakhotia, Ruslan Salakhutdinov, Abdelrahman Mohamed

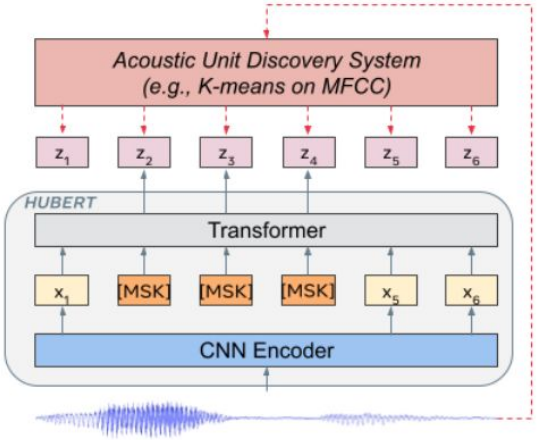


Fig. 1: The HuBERT approach predicts hidden cluster assignments of the masked frames (y_2, y_3, y_4 in the figure) generated by one or more iterations of k-means clustering.

Current Trends: SSL- HuBERT

Main difference from Wav2Vec2:

- **Codebook** (lexicon) is generated using **k-means** on the **embeddings (starts on MFCCs and than modified to intermediate layer 2-3 times during training)**.
- **cross entropy loss**.
- No quantization

We propose to use **acoustic unit discovery models** to provide **frame-level targets**. Let X denote a speech utterance X = [x1, · · · , xT] of T frames. Discovered hidden units are denoted with h(X) = Z = [z1, · · · , zT], where z_t ∈ [C] **is a C-class categorical variable** and **h** is a **clustering model, e.g. k-means.**

HuBERT: Self-Supervised Speech Representation Learning by Masked Prediction of Hidden Units

Wei-Ning Hsu, Benjamin Bolte, Yao-Hung Hubert Tsai, Kushal Lakhotia, Ruslan Salakhutdinov, Abdelrahman Mohamed

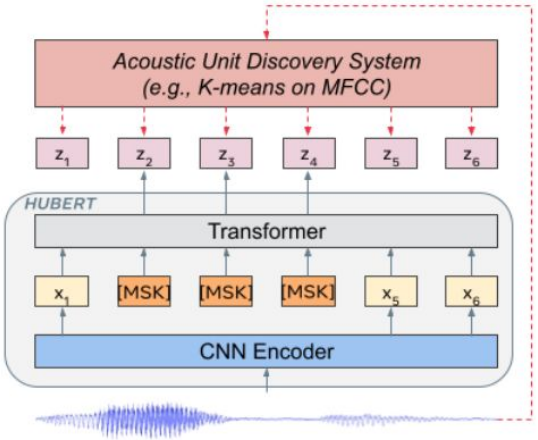


Fig. 1: The HuBERT approach predicts hidden cluster assignments of the masked frames (y_2, y_3, y_4 in the figure) generated by one or more iterations of k-means clustering.

Current Tre

Main difference from Wav2

- **Codebook** (lexicon) i
  k-means on the **emb**
  MFCCs and than mo
  intermediate layer 2
  training).
- **cross entropy loss**.
- No quantization

We propose to use **acoust**
models to provide **frame-l**
denote a speech utterance
T frames. Discovered hidd
denoted with h(X) = Z = [z1,
∈ [C] **is a C-class categori**
is a **clustering model, e.g.**

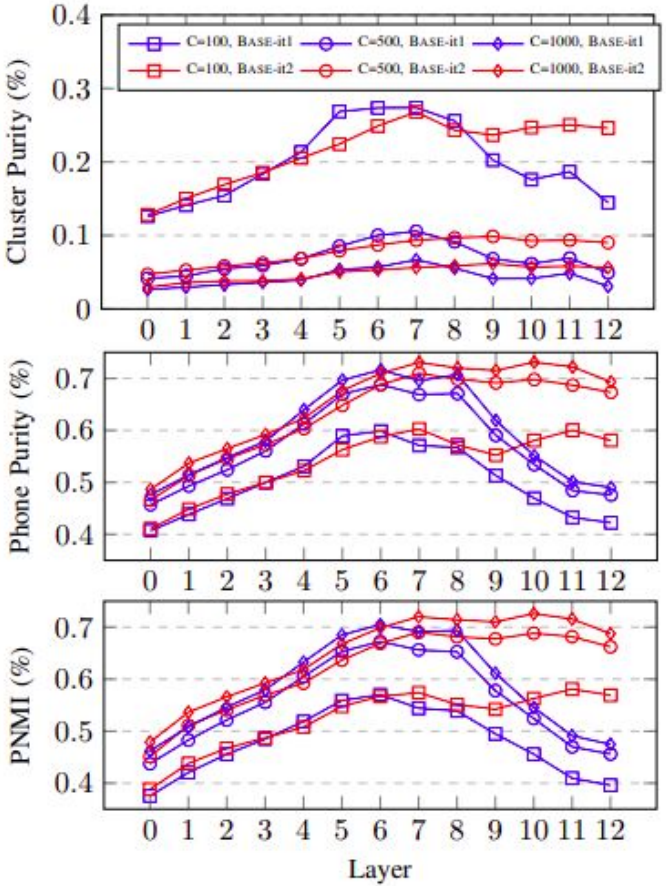
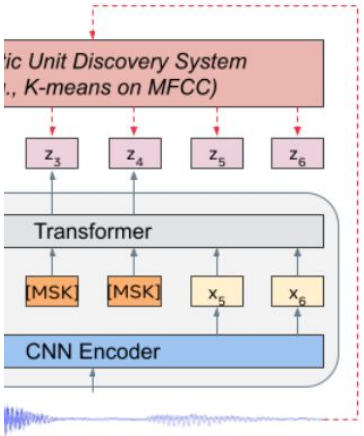


Fig. 2: Quality of the cluster assignments obtained by running k-means clustering on features extracted from each transformer layer of the first and the second iteration BASE HuBERT models.

ised Speech Representation
Prediction of Hidden Units

e, Yao-Hung Hubert Tsai, Kushal Lakhotia,
inov, Abdelrahman Mohamed



approach predicts hidden cluster assign-
frames (y_2, y_3, y_4 in the figure) generated
ions of k-means clustering.

Current Trends: SSL- HuBERT

The convolutional waveform encoder generates a feature sequence at a 20ms framerate for audio sampled at 16kHz (CNN encoder down-sampling factor is 320x). The audio encoded features are then randomly masked as described in Section II-B. The BERT encoder takes as input the masked sequence and outputs a feature sequence $[o_1, \cdots, o_T]$. The distribution over codewords is parameterized with

$$p_f^{(k)}(c \mid \tilde{X}, t) = \frac{\exp(\text{sim}(A^{(k)} o_t, e_c)/\tau)}{\sum_{c'=1}^{C} \exp(\text{sim}(A^{(k)} o_t, e_{c'})/\tau)}, \quad (3)$$

where A is the projection matrix, e_c is the embedding for codeword c, $\text{sim}(\cdot, \cdot)$ computes the cosine similarity between two vectors, and τ scales the logit, which is set to 0.1. When cluster ensembles are used, one projection matrix $A^{(k)}$ is applied for each clustering model k.

After HuBERT pre-training, We use the connectionist temporal classification (CTC) [41] loss for ASR fine-tuning of the whole model weights except the convolutional audio encoder, which remains frozen. The projection layer(s) is removed and replaced with a randomly initialized softmax layer. The CTC target vocabulary includes 26 English characters, a space token, an apostrophe, and a special CTC blank symbol.

HuBERT: Self-Supervised Speech Representation Learning by Masked Prediction of Hidden Units

Wei-Ning Hsu, Benjamin Bolte, Yao-Hung Hubert Tsai, Kushal Lakhotia,
Ruslan Salakhutdinov, Abdelrahman Mohamed

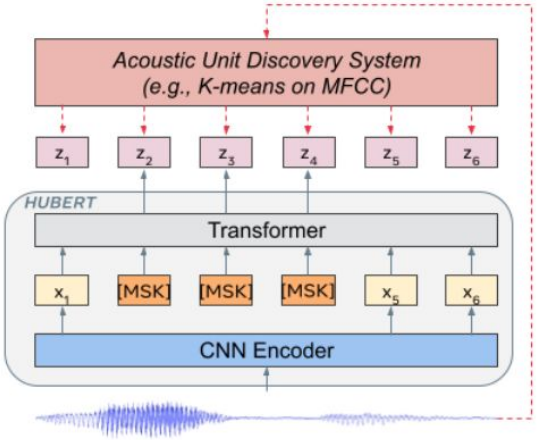


Fig. 1: The HuBERT approach predicts hidden cluster assignments of the masked frames (y_2, y_3, y_4 in the figure) generated by one or more iterations of k-means clustering.

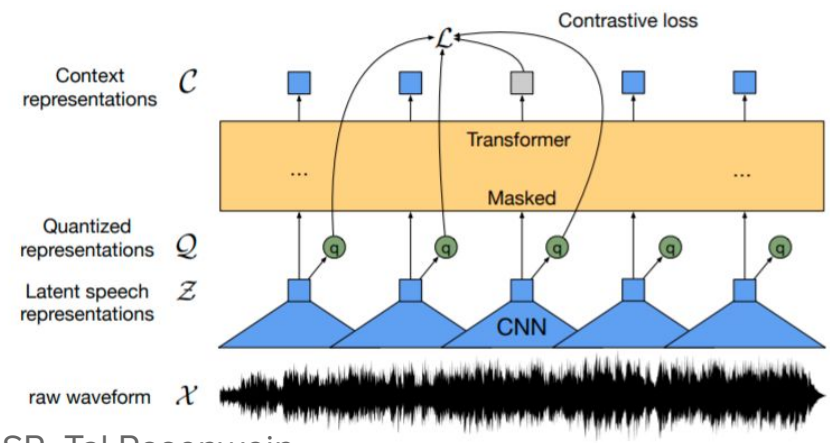
Current Trends: SSL- Pre training

wav2vec 2.0: A Framework for Self-Supervised Learning of Speech Representations

Alexei Baevski Henry Zhou Abdelrahman Mohamed Michael Auli

{abaevski,henryzhou7,abdo,michaelauli}@fb.com

Facebook AI



HuBERT: Self-Supervised Speech Representation Learning by Masked Prediction of Hidden Units

Wei-Ning Hsu, Benjamin Bolte, Yao-Hung Hubert Tsai, Kushal Lakhotia, Ruslan Salakhutdinov, Abdelrahman Mohamed

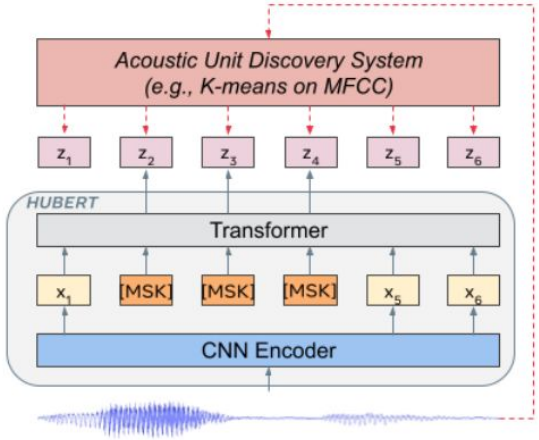
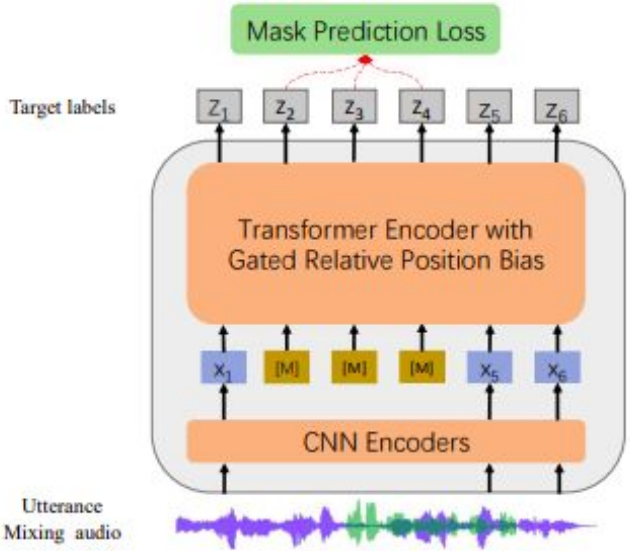


Fig. 1: The HuBERT approach predicts hidden cluster assignments of the masked frames (y_2, y_3, y_4 in the figure) generated by one or more iterations of k-means clustering.

Current Trends: SSL- Pre training

WavLM: Large-Scale Self-Supervised Pre-Training for Full Stack Speech Processing

Sanyuan Chen*, Chengyi Wang*, Zhengyang Chen*, Yu Wu*, Shujie Liu, Zhuo Chen, Jinyu Li, Naoyuki Kanda, Takuya Yoshioka, Xiong Xiao, Jian Wu, Long Zhou, Shuo Ren, Yanmin Qian, Yao Qian, Jian Wu, Michael Zeng, Xiangzhan Yu, Furu Wei



W2V-BERT: COMBINING CONTRASTIVE LEARNING AND MASKED LANGUAGE MODELING FOR SELF-SUPERVISED SPEECH PRE-TRAINING

Yu-An Chung[1,2*], *Yu Zhang*[2], *Wei Han*[2], *Chung-Cheng Chiu*[2], *James Qin*[2], *Ruoming Pang*[2], *Yonghui Wu*[2]

[1]MIT Computer Science and Artificial Intelligence Laboratory
[2]Google Brain

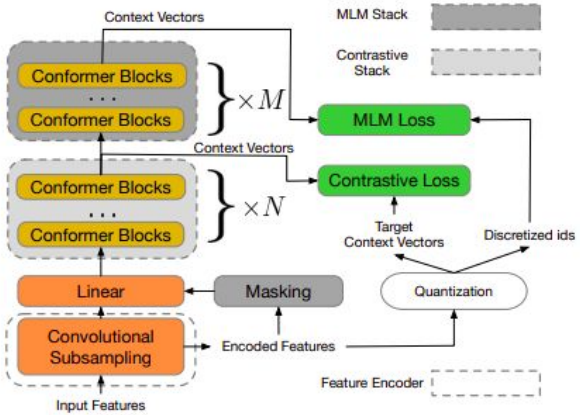


Fig. 1: Illustration of the w2v-BERT pre-training framework. w2v-BERT is composed of a feature encoder, a contrastive module, and a masked language modeling (MLM) module, where the latter two are both a stack of conformer blocks. N and M denote the number of conformer blocks in the two modules, respectively.

Current Trends: Whisper

Robust Speech Recognition via Large-Scale Weak Supervision

Alec Radford [*1] Jong Wook Kim [*1] Tao Xu [1] Greg Brockman [1] Christine McLeavey [1] Ilya Sutskever [1]

Abstract

We study the capabilities of speech processing systems trained simply to predict large amounts of transcripts of audio on the internet. When scaled to 680,000 hours of multilingual and multitask supervision, the resulting models generalize well to standard benchmarks and are often competitive with prior fully supervised results but in a zero-shot transfer setting without the need for any fine-tuning. When compared to humans, the models approach their accuracy and robustness. We are releasing models and inference code to serve as a foundation for further work on robust speech processing.

Current Trends: Whisper

Robust Speech Recognition via Large-Scale Weak Supervision

Alec Radford [*1] Jong Wook Kim [*1] Tao Xu [1] Greg Brockman [1] Christine McLeavey [1] Ilya Sutskever [1]

- Encoder-Decoder architecture (Scale models)
- Data curation - Scale to **680,000 hours.** Human generated (apply filters on machine-generated transcripts).
- **multilingual** and **multitask** supervision
  - **(117K hours cover 96 languages)**
- **Competitive** with prior fully supervised results but **without the need for any finetuning**.
- **No text normalization** (hence no inverse text normalization)
- GPT2 tokenizer
- Use 30s audio files, **add context** to the decoder.
- **Hallucinations problems**

Curre

Ale

- Encoder-De
- Data curat … pply filters on machine-ge
- **multilingua**
  - **(117K**
- **Competitiv** … **any finetuning**.
- No text nor
- GPT2 toker
- Use 30s au
- Hallucinatic

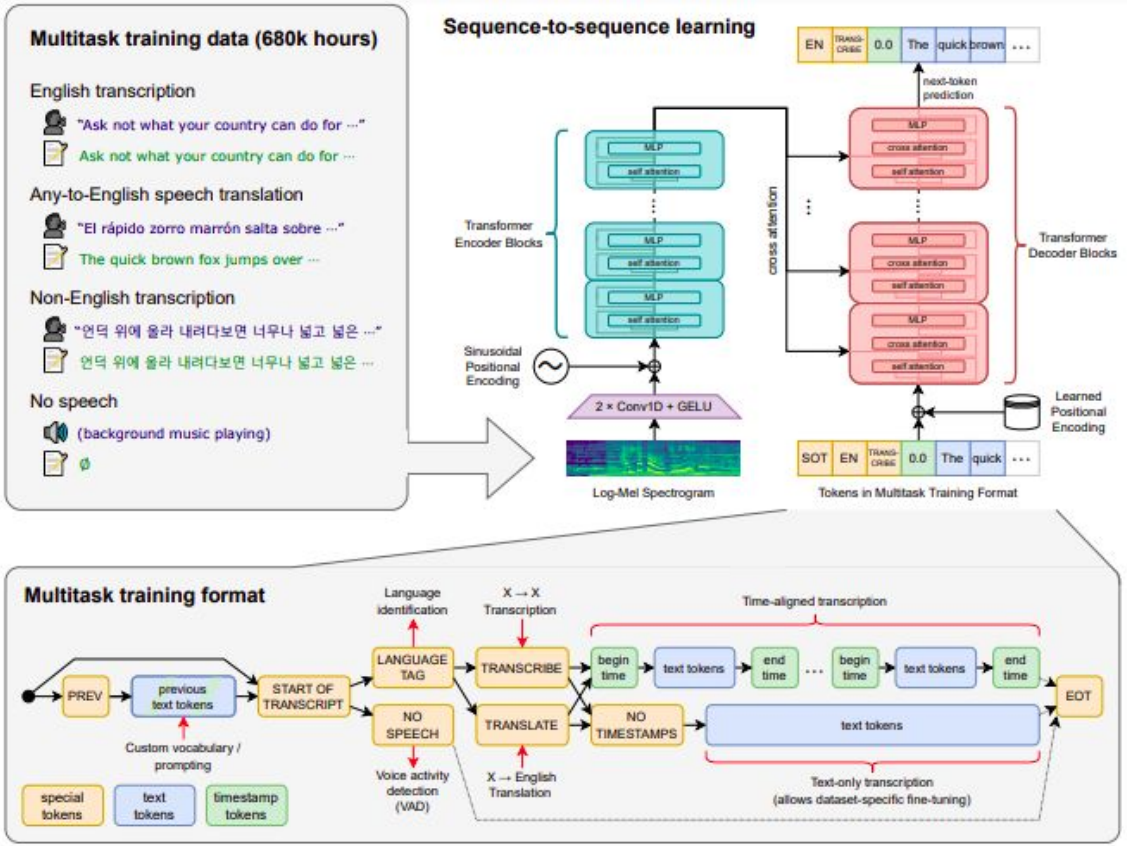


Figure 1. **Overview of our approach.** A sequence-to-sequence Transformer model is trained on many different speech processing tasks, including multilingual speech recognition, speech translation, spoken language identification, and voice activity detection. All of these tasks are jointly represented as a sequence of tokens to be predicted by the decoder, allowing for a single model to replace many different stages of a traditional speech processing pipeline. The multitask training format uses a set of special tokens that serve as task specifiers or classification targets, as further explained in Section 2.3.

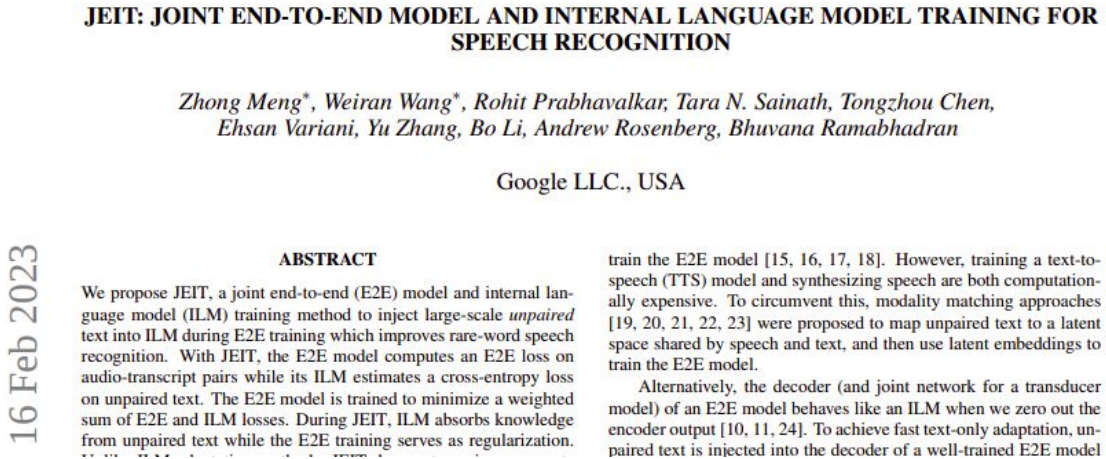
Current Trends: Whisper

Robust Speech Recognition via Large-Scale Weak Supervision

Alec Radford [*1] Jong Wook Kim [*1] Tao Xu [1] Greg Brockman [1] Christine McLeavey [1] Ilya Sutskever [1]

- Encoder-Decoder architecture (Scale models)
- Data curation - Scale to **680,000 hours.** Human generated (apply filters on machine-generated transcripts).
- **multilingual** and **multitask** supervision
  - **(117K hours cover 96 languages)**
- **Competitive** with prior fully supervised results but **without the need for any finetuning**.
- **No text normalization** (hence no inverse text normalization)
- GPT2 tokenizer
- Use 30s audio files, **add context** to the decoder.
- **Hallucinations problems**

Current Trends: JEIT



JEIT: JOINT END-TO-END MODEL AND INTERNAL LANGUAGE MODEL TRAINING FOR SPEECH RECOGNITION

Zhong Meng, Weiran Wang*, Rohit Prabhavalkar, Tara N. Sainath, Tongzhou Chen, Ehsan Variani, Yu Zhang, Bo Li, Andrew Rosenberg, Bhuvana Ramabhadran*

Google LLC., USA

ABSTRACT

We propose JEIT, a joint end-to-end (E2E) model and internal language model (ILM) training method to inject large-scale *unpaired* text into ILM during E2E training which improves rare-word speech recognition. With JEIT, the E2E model computes an E2E loss on audio-transcript pairs while its ILM estimates a cross-entropy loss on unpaired text. The E2E model is trained to minimize a weighted sum of E2E and ILM losses. During JEIT, ILM absorbs knowledge from unpaired text while the E2E training serves as regularization.

train the E2E model [15, 16, 17, 18]. However, training a text-to-speech (TTS) model and synthesizing speech are both computationally expensive. To circumvent this, modality matching approaches [19, 20, 21, 22, 23] were proposed to map unpaired text to a latent space shared by speech and text, and then use latent embeddings to train the E2E model.

Alternatively, the decoder (and joint network for a transducer model) of an E2E model behaves like an ILM when we zero out the encoder output [10, 11, 24]. To achieve fast text-only adaptation, unpaired text is injected into the decoder of a well-trained E2E model

- Improve internal LM
- Training AM & ILM: regular training
- Training solely the ILM: Zeroing acoustic model and feeding the only text (cross entropy)

Current Trends: Distillation (Teacher-Student)

IMPROVING STREAMING AUTOMATIC SPEECH RECOGNITION WITH NON-STREAMING MODEL DISTILLATION ON UNSUPERVISED DATA- Google 2021

We propose a novel and effective learning method by leveraging a non-streaming ASR model as a teacher to generate transcripts (**pseudo labels**) on an arbitrarily large data set, which is then used to distill knowledge into streaming ASR models. This way, we scale the training of streaming models to up to **3 million hours** of YouTube audio

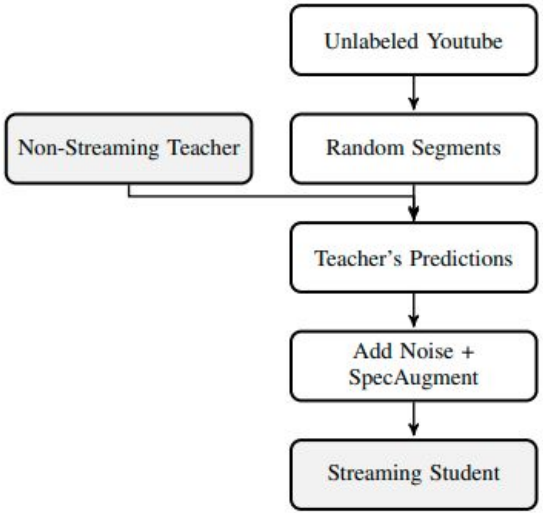


Fig. 1. Our method trains a streaming model, learning from the predictions of a powerful non-streaming teacher model on large-scale unlabeled data via a teacher-student training framework. See Sect. 2.2 for more details.

Current Trends: Distillation (Teacher-Student)

IMPROVING STREAMING AUTOMATIC SPEECH RECOGNITION WITH NON-STREAMING MODEL DISTILLATION ON UNSUPERVISED DATA- Google 2021

In this paper, we propose a new approach to train end-to-end streaming models from unsupervised data. Our approach can be divided into three steps:

- We employ the state-of-the-art full-context model as a teacher model.
- We convert unlabeled audio sequences into random segments and transcribe them using the full-context teacher model.
- We use the waveforms and their predicted transcripts in a noisy student learning framework [14, 15, 16, 17].
- Teacher (non-streaming conformer transducer with **4K word-piece**) has 179M params and student (streaming RNN-T) has 122M params.

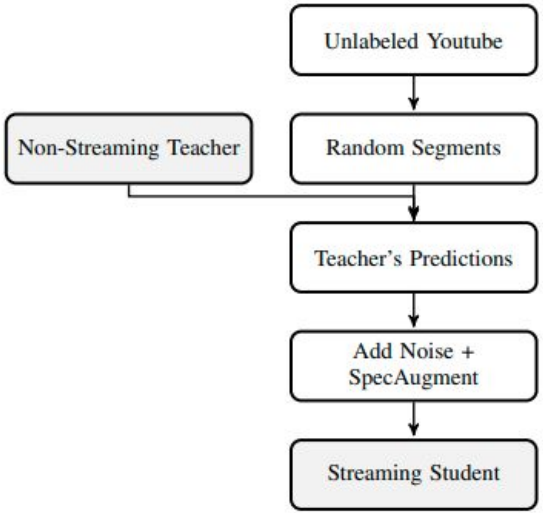


Fig. 1. Our method trains a streaming model, learning from the predictions of a powerful non-streaming teacher model on large-scale unlabeled data via a teacher-student training framework. See Sect. 2.2 for more details.

Current Trends: Distillation (Teacher-Student)

IMPROVING STREAMING AUTOMATIC SPEECH RECOGNITION WITH NON-STREAMING MODEL DISTILLATION ON UNSUPERVISED DATA- Google 2021

Fig. 1 illustrates our idea. Given unlabeled YouTube data, we convert them into segments with random lengths. Such random segments do not require an alignment model while still providing good samples for ASR training. A non-streaming teacher model is then used to transcribe these random segments. These predictions can be viewed as **pseudo labels**. The random segments with these pseudo labels will be used to train a student model. Following [14], we augment the inputs with noise when training the student network, to make the student model more robust.

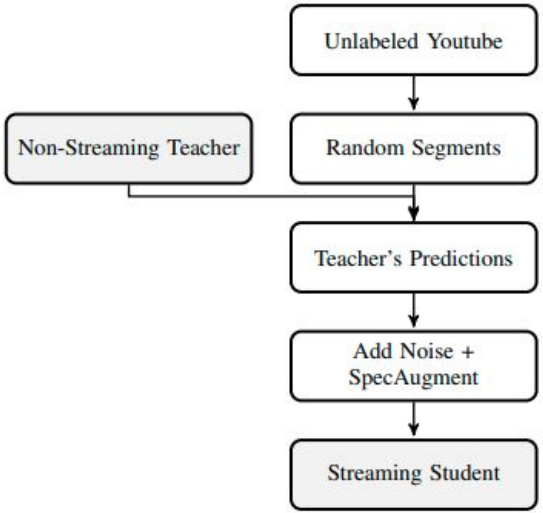


Fig. 1. Our method trains a streaming model, learning from the predictions of a powerful non-streaming teacher model on large-scale unlabeled data via a teacher-student training framework. See Sect. 2.2 for more details.

Current Trends: Large Scale (Models & Data)

IMPROVING STREAMING AUTOMATIC SPEECH RECOGNITION WITH NON-STREAMING MODEL DISTILLATION ON UNSUPERVISED DATA

Thibault Doutre, Wei Han*, Min Ma, Zhiyun Lu, Chung-Cheng Chiu, Ruoming Pang, Arun Narayanan, Ananya Misra, Yu Zhang, Liangliang Cao*

Google Inc., USA

{doutre,weihan,llcao}@google.com

3M Hours

BigSSL: Exploring the Frontier of Large-Scale Semi-Supervised Learning for Automatic Speech Recognition

Yu Zhang*, Daniel S. Park*, Wei Han*,
James Qin, Anmol Gulati, Joel Shor, Aren Jansen,
Yuanzhong Xu, Yanping Huang, Shibo Wang, Zongwei Zhou,
Bo Li, Min Ma, William Chan, Jiahui Yu, Yongqiang Wang,
Liangliang Cao, Khe Chai Sim, Bhuvana Ramabhadran,
Tara N. Sainath, Françoise Beaufays, Zhifeng Chen, Quoc V. Le, Chung-Cheng Chiu,
Ruoming Pang† and Yonghui Wu

~1M Hours

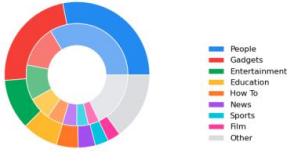


Fig. 4: Video categories by length (outer) and number (inner).

TABLE I: Conformer model parameters.

Model	# Params (B)	# Layers	Dimension	Att. Heads
Conformer XL	0.6	24	1024	8
Conformer XXL	1.0	42	1024	8
Conformer G	8.0	36	3072	16

SCALING ASR IMPROVES ZERO AND FEW SHOT LEARNING

Alex Xiao, Weiyi Zheng*, Gil Keren, Duc Le, Frank Zhang, Christian Fuegen Ozlem Kalinli, Yatharth Saraf, Abdelrahman Mohamed*

Facebook AI

4M -> 1.3M Hours

Data Source	Transcriber	Hours	Hours After Augmentation
LibriSpeech [18]	Human	960	5760
Common Voice [19]	Human	500	3000
Libri-Light [20]	Model	60000	360000
Fisher [21]	Human	1960	11760
Assistant*	Human	12600	41400
Conversational*	Human	780	6600
Calling Names	TTS	640	3840
Dictation*	Model	880	7920
Portal †	Human	1350	8100
Video †	Human	18000	108000
Portal †	Model	4800	28800
Video †	Model	4009400	4009400

Table 1. Our 4.5M hour dataset consists of 10 sources. Data sources marked with * are collected through third-party vendors. Those marked with † are collected from Facebook products.

Parameters	Hidden Size	Layers	Attention Heads
100M	512	36	8
1B	1152	60	16
10B	3072	90	48

Table 2. Hyper-parameters for our Transformer encoders.

What's Next: Unsupervised ASR

ASR has seen rapid improvement over the last few years due to advances in model architectures, semi-supervised learning and self-supervised learning. However, all of these techniques require **transcribed speech data** which is **not available** for the **vast majority** of the nearly **7,000 languages** of the world. As a result, speech recognition technology is only **available** for about **125 different languages**. On the other hand, **humans learn** a lot about speech **simply by listening to others** around them and **without explicit supervision** .

https://www.salon.com/2017/04/23/what-you-do-not-know-about-the-holy-grail/

What's Next: Unsupervised ASR

Unsupervised Speech Recognition

Alexei Baevski[△], Wei-Ning Hsu[△], Alexis Conneau[□*], Michael Auli[△]
[△] Facebook AI [□] Google AI

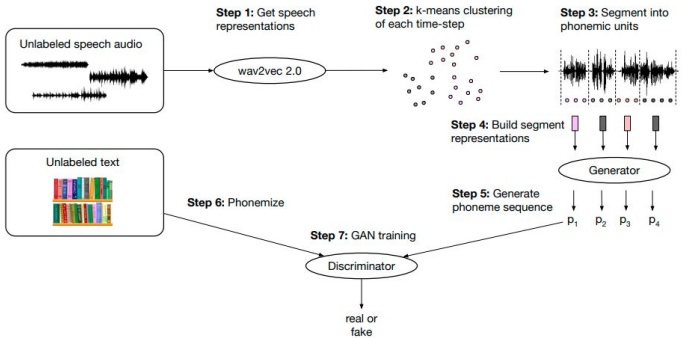


Figure 1: Illustration of wav2vec Unsupervised: we learn self-supervised representations with wav2vec 2.0 on unlabeled speech audio (Step 1), then identify clusters in the representations with k-means (Step 2) to segment the audio data (Step 3). Next, we build segment representations by mean pooling the wav2vec 2.0 representations, performing PCA and a second mean pooling step between adjacent segments (Step 4). This is input to the generator which outputs a phoneme sequence (Step 5) that is fed to the discriminator, similar to phonemized unlabeled text (Step 6) to perform adversarial training (Step 7).

ASR

https://arxiv.org/pdf/2105.11084.pdf

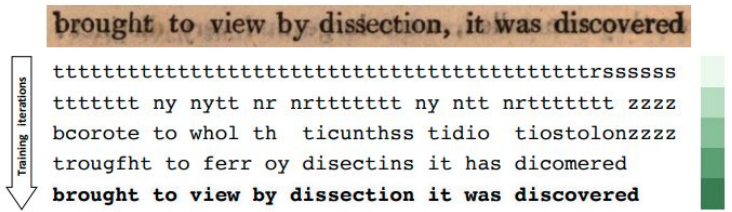
Learning to Read by Spelling

Towards Unsupervised Text Recognition

Ankush Gupta
Visual Geometry Group
University of Oxford
ankush@robots.ox.ac.uk

Andrea Vedaldi
Visual Geometry Group
University of Oxford
vedaldi@robots.ox.ac.uk

Andrew Zisserman
Visual Geometry Group
University of Oxford
az@robots.ox.ac.uk



OCR

https://arxiv.org/pdf/1809.08675.pdf

What's Next: Unsupervised ASR

We introduce a framework for unsupervised learning of speech recognition models. Wav2vec-U, or wav2vec Unsupervised, **leverages self-supervised representations from wav2vec 2.0** (Baevski et al., 2020c) to embed the speech audio and to segment the audio into units with a simple k-means clustering method. We find that the quality of the audio representations is key to the success of unsupervised speech recognition.

... We learn a **mapping** between **segments and phonemes** using **adversarial training** and enable the algorithm to label segments as silences. We also introduce an unsupervised cross-validation metric to enable model development **without labeled development data**. Our unsupervised speech recognition model, the generator, is very lightweight: it consists of a single temporal convolution comprising only about 90k parameters to which we input frozen wav2vec 2.0 representations.

What's Next: Unsupervised ASR

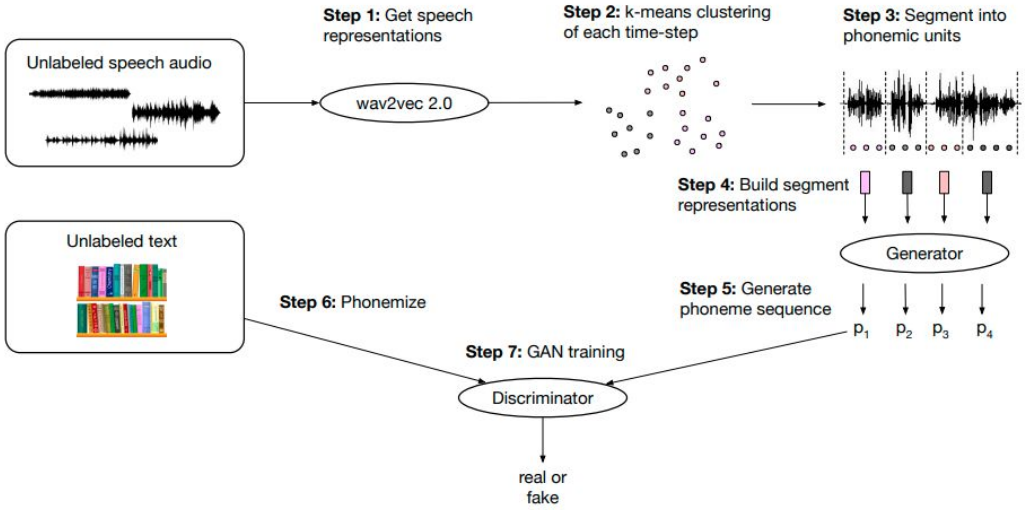


Figure 1: Illustration of wav2vec Unsupervised: we learn self-supervised representations with wav2vec 2.0 on unlabeled speech audio (Step 1), then identify clusters in the representations with k-means (Step 2) to segment the audio data (Step 3). Next, we build segment representations by mean pooling the wav2vec 2.0 representations, performing PCA and a second mean pooling step between adjacent segments (Step 4). This is input to the generator which outputs a phoneme sequence (Step 5) that is fed to the discriminator, similar to phonemized unlabeled text (Step 6) to perform adversarial training (Step 7).

What's Next: Multi-Modal shared embeddings

SpeechT5: Unified-Modal Encoder-Decoder Pre-Training for Spoken Language Processing

Junyi Ao[1,2,*], Rui Wang[3,*], Long Zhou[4,*], Chengyi Wang[4], Shuo Ren[4], Yu Wu[4], Shujie Liu[4], Tom Ko[1], Qing Li[2], Yu Zhang[1,5], Zhihua Wei[3], Yao Qian[4], Jinyu Li[4], Furu Wei[4]

[1]Department of Computer Science and Engineering, Southern University of Science and Technology
[2]Department of Computing, The Hong Kong Polytechnic University
[3]Department of Computer Science and Technology, Tongji University
[4]Microsoft
[5]Peng Cheng Laboratory

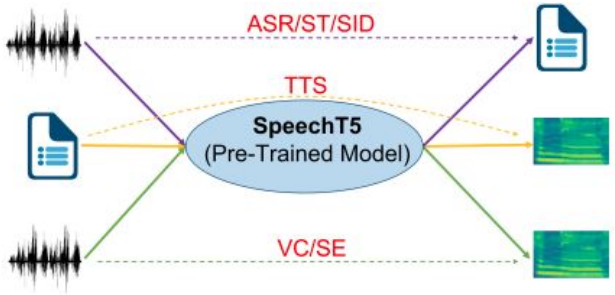


Figure 1: An illustration of the SpeechT5 framework, which treats spoken language processing tasks as a speech/text to speech/text format, including automatic speech recognition (ASR), speech translation (ST), speech identification (SID), text to speech (TTS), voice conversion (VC), and speech enhancement (SE).

VATT: Transformers for Multimodal Self-Supervised Learning from Raw Video, Audio and Text

Hassan Akbari* Columbia University ha2436@columbia.edu
Liangzhe Yuan Google lzyuan@google.com
Rui Qian* Cornell University rq49@cornell.edu
Wei-Hong Chuang Google whchuang@google.com
Shih-Fu Chang Columbia University sc250@columbia.edu
Yin Cui Google yincui@google.com
Boqing Gong Google bgong@google.com

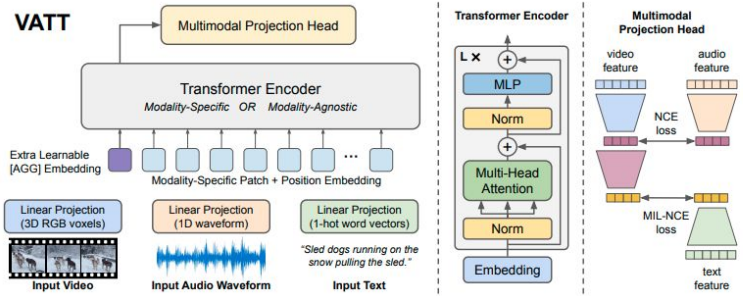


Figure 1: **Overview of the VATT architecture and the self-supervised, multimodal learning strategy**. VATT linearly projects each modality into a feature vector and feeds it into a Transformer encoder. We define a semantically hierarchical common space to account for the granularity of different modalities and employ the Noise Contrastive Estimation (NCE) to train the model.

ASR, Tal Rosenwein

What's Next: Multi-Modal shared embeddings

SpeechT5: Unified-Modal Encoder-Decoder Pre-Training for Spoken Language Processing

Junyi Ao[1,2,*], Rui Wang[3,*], Long Zhou[4,*], Chengyi Wang[4], Shuo Ren[4], Yu Wu[4], Shujie Liu[4], Tom Ko[1], Qing Li[2], Yu Zhang[1,5], Zhihua Wei[3], Yao Qian[4], Jinyu Li[4], Furu Wei[4]

[1]Department of Computer Science and Engineering, Southern University of Science and Technology
[2]Department of Computing, The Hong Kong Polytechnic University
[3]Department of Computer Science and Technology, Tongji University
[4]Microsoft
[5]Peng Cheng Laboratory

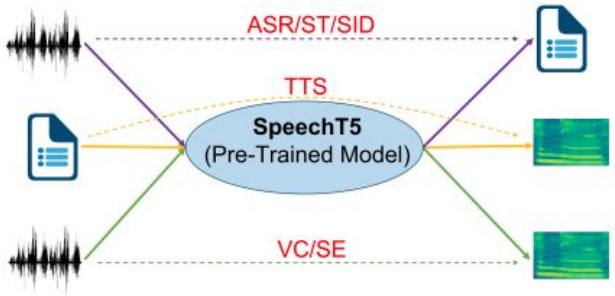


Figure 1: An illustration of the SpeechT5 framework, which treats spoken language processing tasks as a speech/text to speech/text format, including automatic speech recognition (ASR), speech translation (ST), speech identification (SID), text to speech (TTS), voice conversion (VC), and speech enhancement (SE).

we propose a unified-modal SpeechT5 framework that explores the encoder-decoder pre-training for **self-supervised speech/text representation learning**. The SpeechT5 framework consists of a **shared encoder-decoder network** and six modal-specific (speech/text) pre/post-nets. After preprocessing the input speech/text through the pre-nets, the shared encoder-decoder network models the seq-to-seq transformation, and then the post-nets generate the output in the speech/text modality based on the output of the decoder. Leveraging largescale unlabeled speech and text data, **we pretrain SpeechT5 to learn a unified-modal representation, hoping to improve the modeling capability for both speech and text.** Extensive evaluations show the **superiority** of the proposed SpeechT5 framework **on a wide variety of spoken language processing tasks**, including automatic speech recognition, speech synthesis, speech translation, voice conversion, speech enhancement, and speaker identification.

different modalities and employ the Noise Contrastive Estimation (NCE) to train the model.

ASR, Tal Rosenwein

ASR Questions?

Tal Rosenwein