End-to-End ASR

Presented by Aku Rouhe

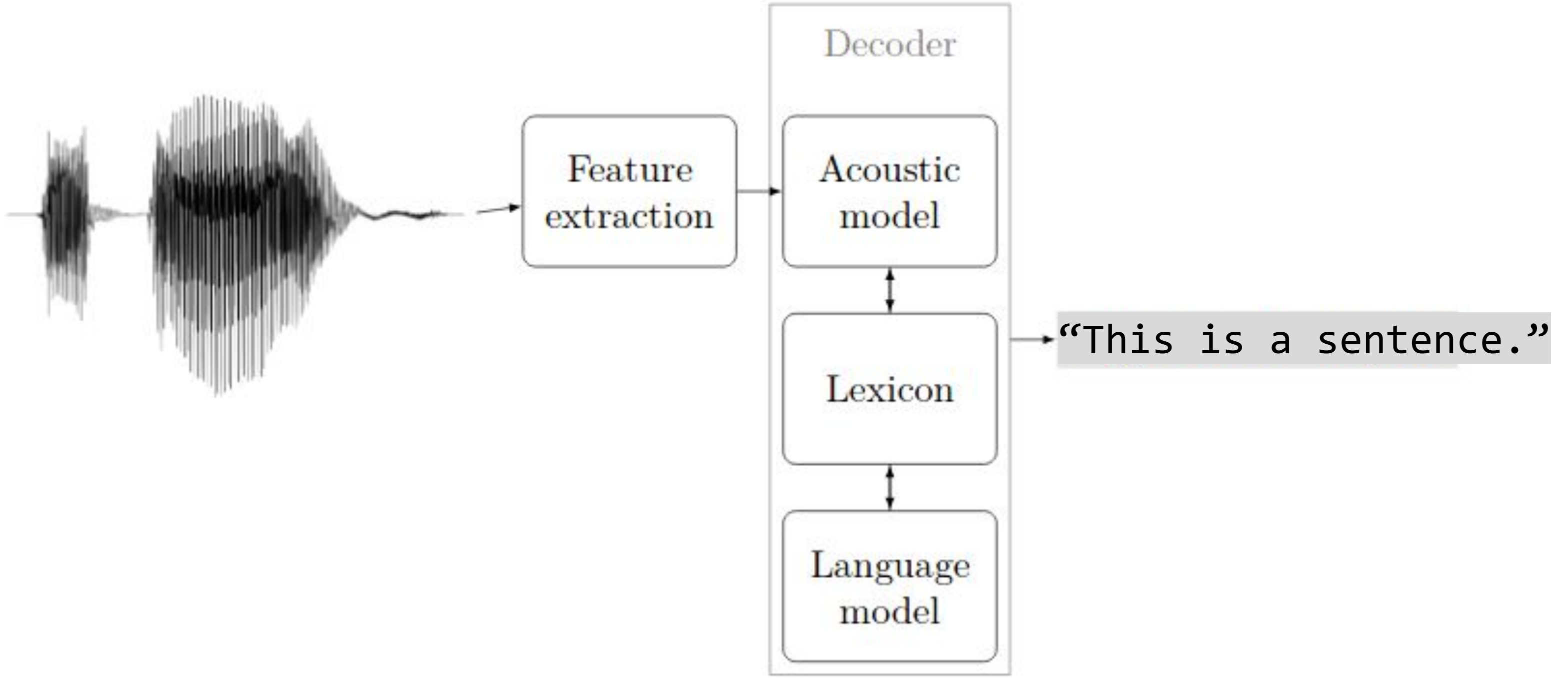
Aalto University
School of Electrical
Engineering

Isn’t all ASR end-to-end?

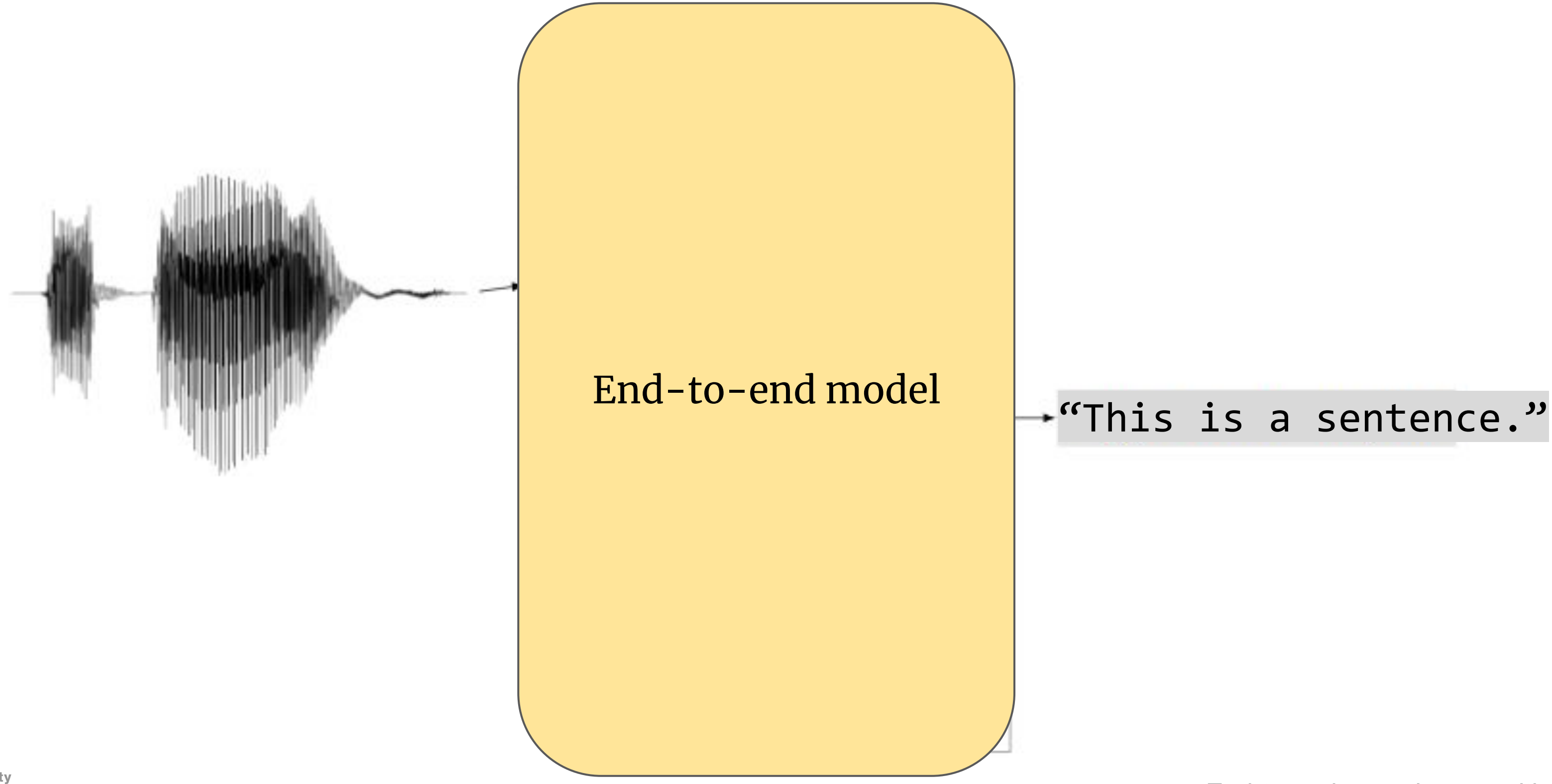
“This is a sentence.”

End-to-End is a Vague Umbrella term

HMM-system: Multiple models



E2E-model: Directly from audio to text



Simplify ASR

GMM

DNN

HMM

N-gram language model

Pronunciation model

Neural network language model

End-to-end model

A look at search spaces

Multimodel: $$\arg_{w}\max\ p(\mathbf{O} \mid \mathbf{s})p(\mathbf{s} \mid \mathbf{w})p(\mathbf{w})$$

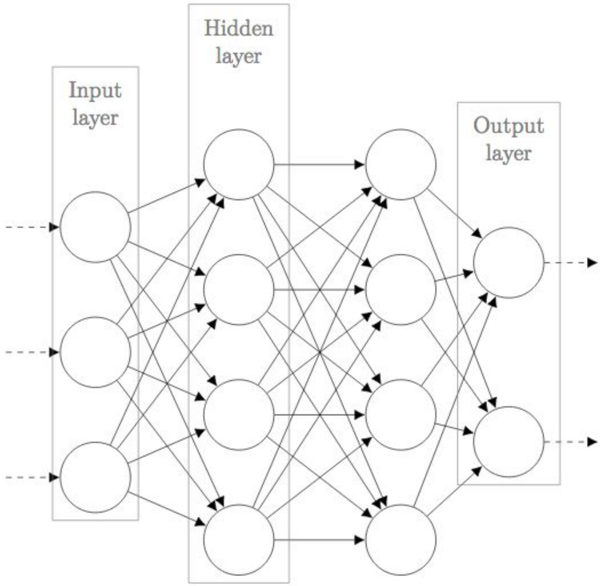
End-to-End: $$\arg_{w}\max\ p(\mathbf{w} \mid \mathbf{O})$$

Joint training, Joint decoding

- *Joint decoding*: Use all submodels together - before pruning
  - e.g. Decoding algorithm combines $p(\mathbf{O} \mid \mathbf{s})$, $p(\mathbf{s} \mid \mathbf{w})$, and $p(\mathbf{w})$
- *Joint training*: Train all submodels together - avoid suboptimization
  - e.g. One global training criterion

How to model p(**w**|**O**) directly?

Use a big neural network



Is End-to-End better?

- Not necessarily in terms of WER
- End-to-End systems can more easily run on e.g. a mobile phone

Table of contents today:

- Connectionist Temporal Classification
- Neural Transducer

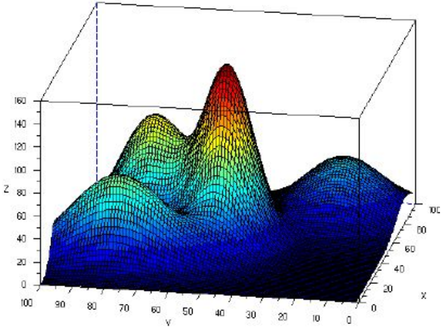
  BREAK
- Attention-based Encoder-Decoder

Kahoot

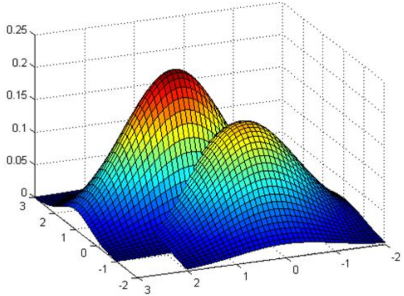
Background from HMM Acoustic Models

GMM

$p(\mathbf{x}_1 | y_t\text{=/ih/})$

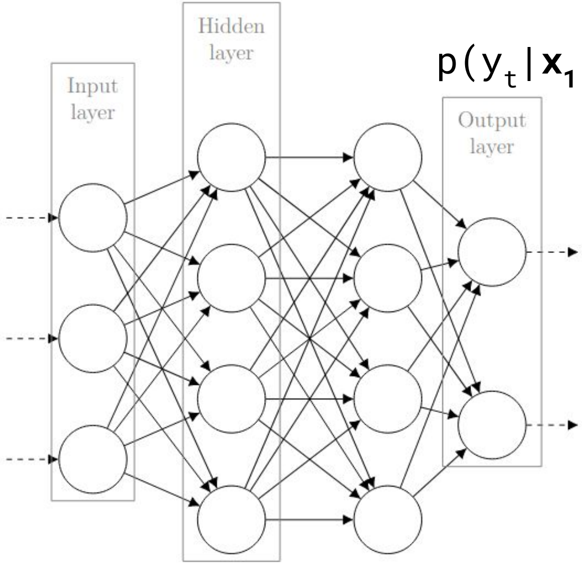


$p(\mathbf{x}_1 | y_t\text{=/z/})$



DNN

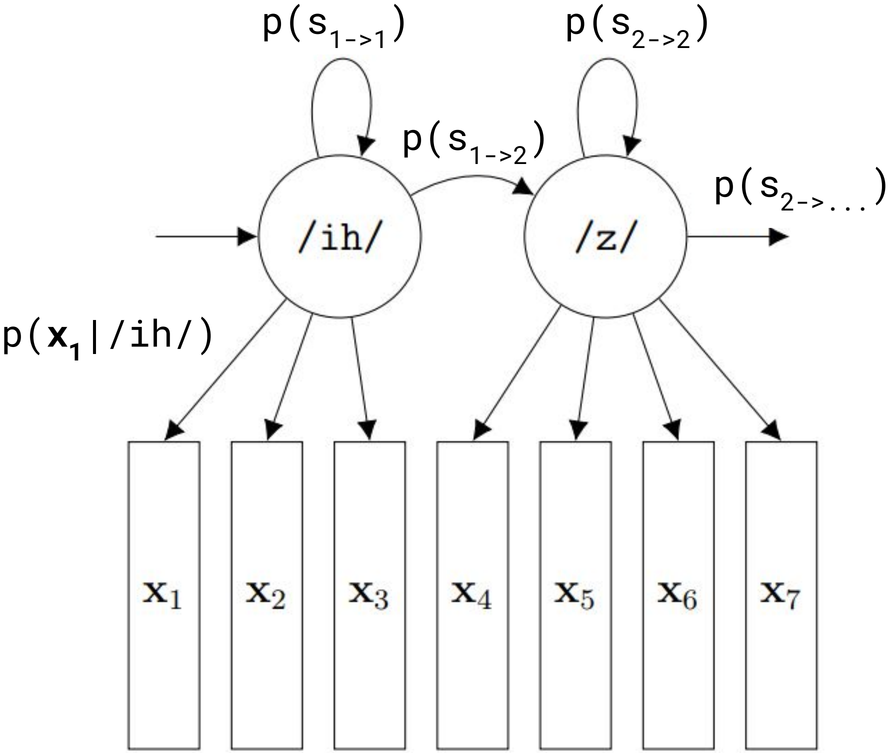
$p(y_t | \mathbf{x}_1)$



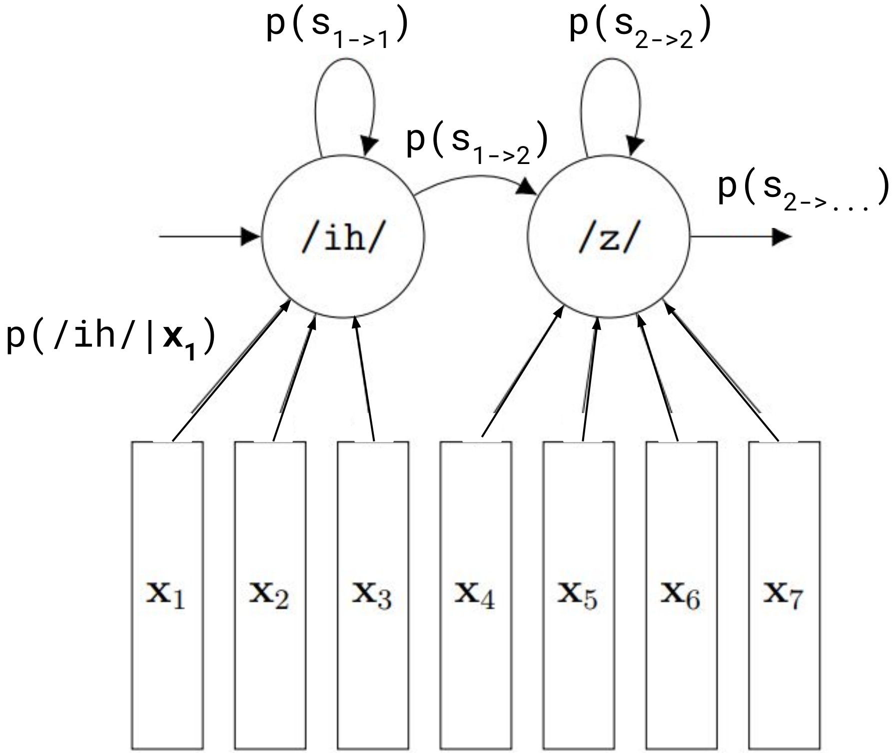
$p(y_t\text{=/ih/} | \mathbf{x}_1)$

$p(y_t\text{=/z/} | \mathbf{x}_1)$

HMM / GMM

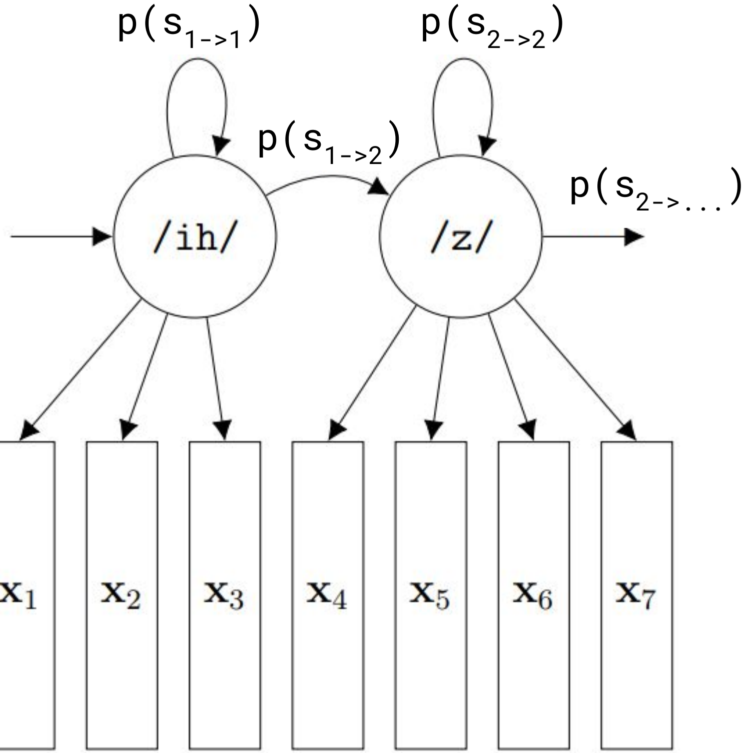


HMM / DNN

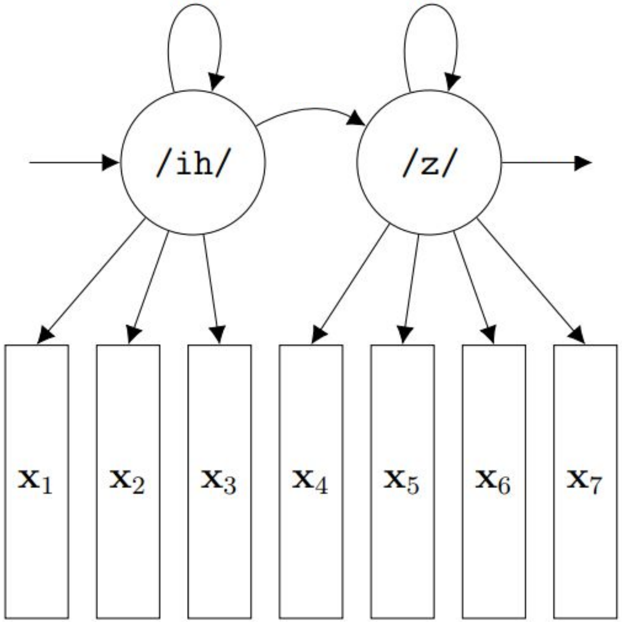


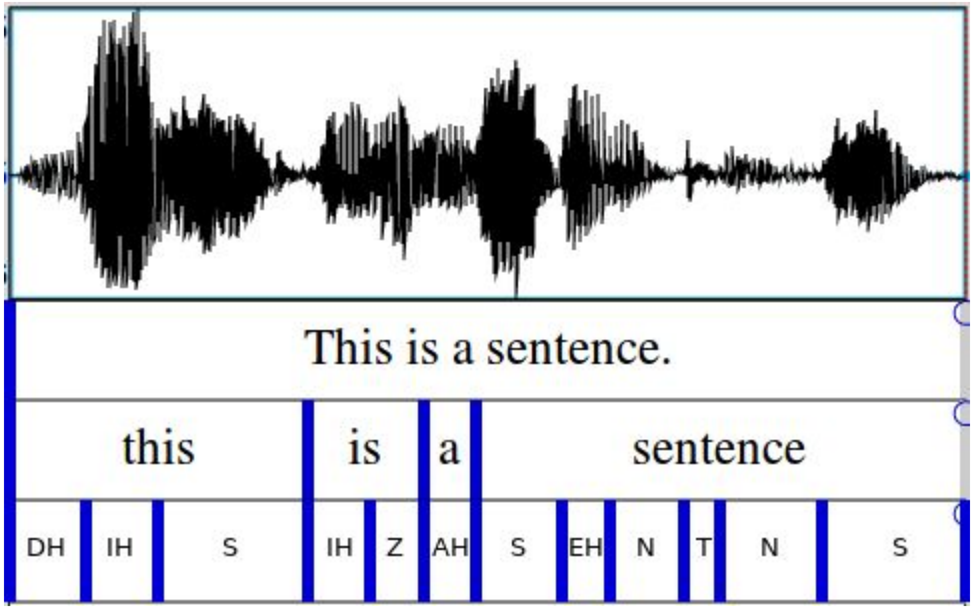
HMM / DNN

$$\frac{p(\mathbf{x}_1|/ih/)}{p(\mathbf{x}_1)} = \frac{p(/ih/|\mathbf{x}_1)}{p(/ih/)}$$

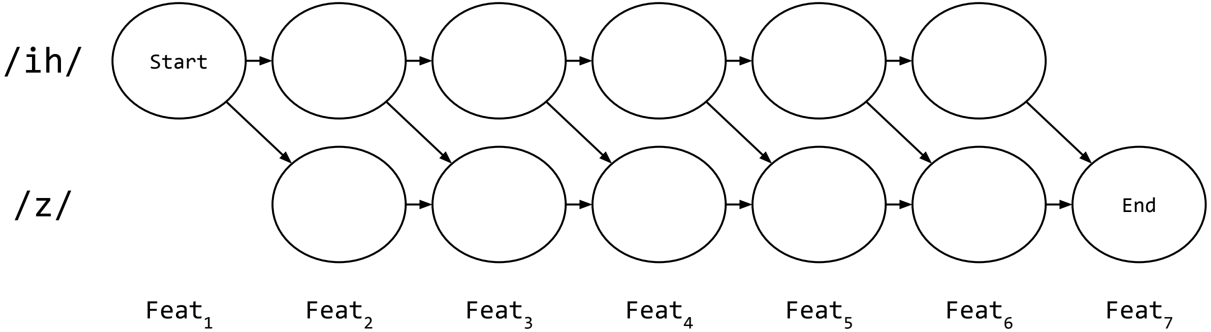


HMM Alignment

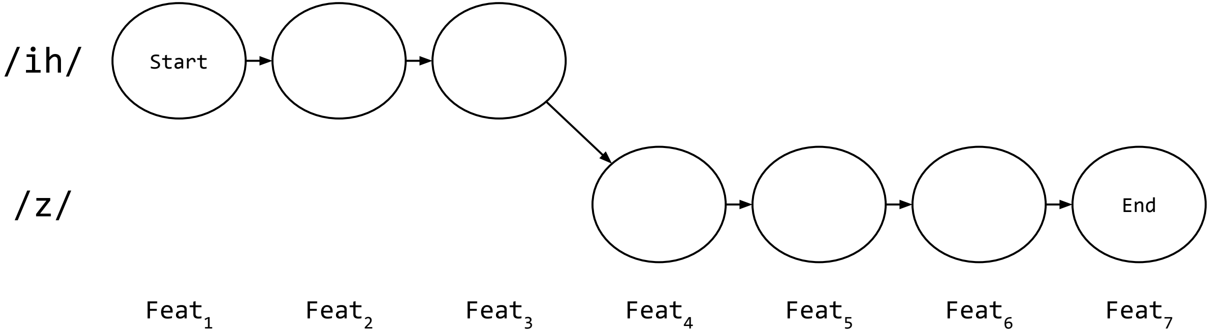




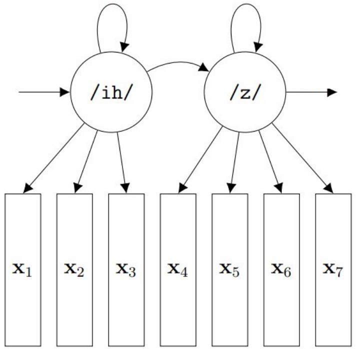
Full-Sum Training (Forward-Backward)

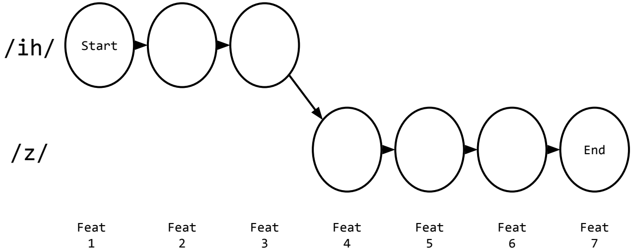


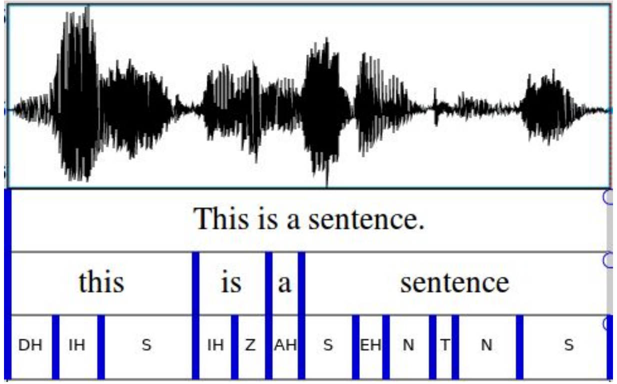
Viterbi



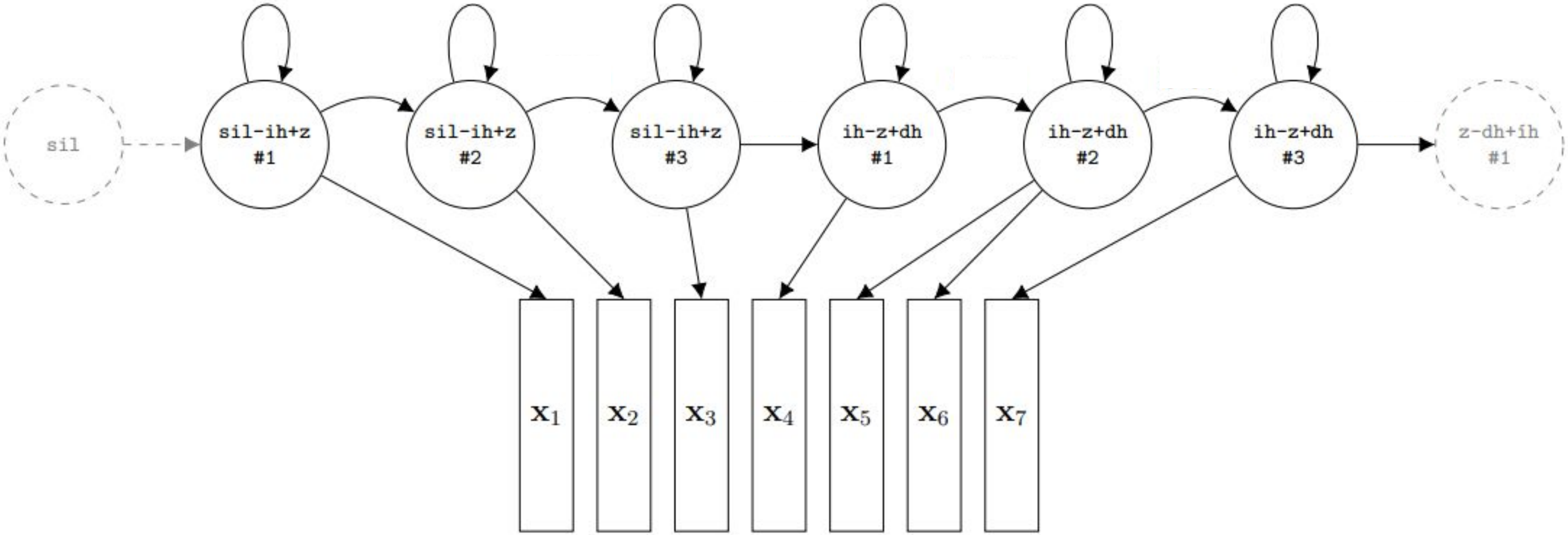
HMM Alignment





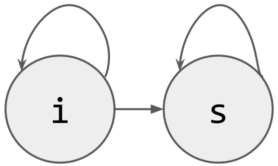


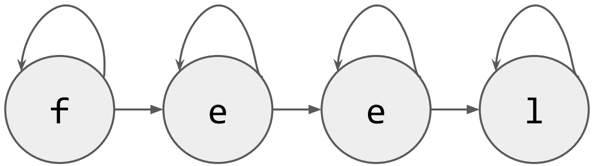
Triphone Tristate HMM



A simpler HMM / DNN system?

- Full sum training doesn't need existing alignments
- What about tristate triphone HMMs and the state tying they need - could we do without it?
- What about phone units - could do without them as well, and just use characters?





Connectionist Temporal Classification

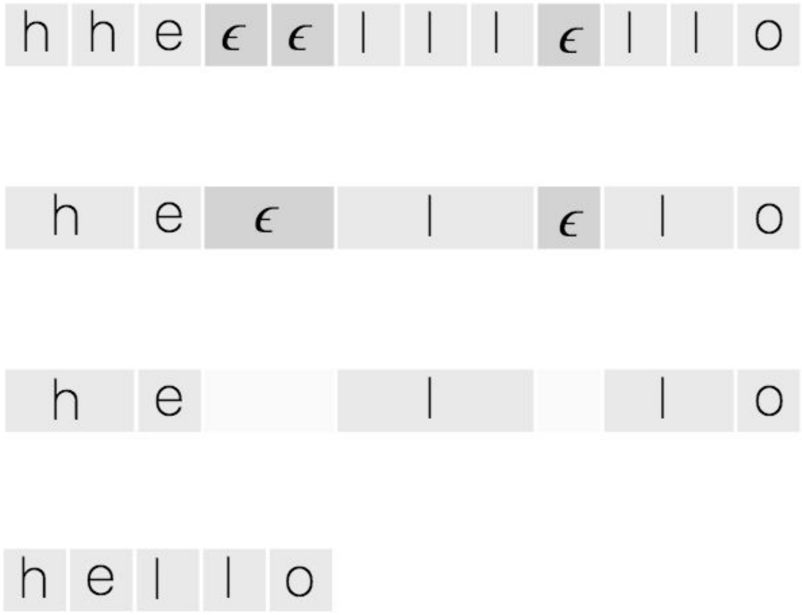
Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks

Alex Graves, Santiago Fernández, Faustino Gomez, and Jürgen Schmidhuber

2006

In Proceedings of the 23rd international conference on Machine learning (ICML)

CTC output



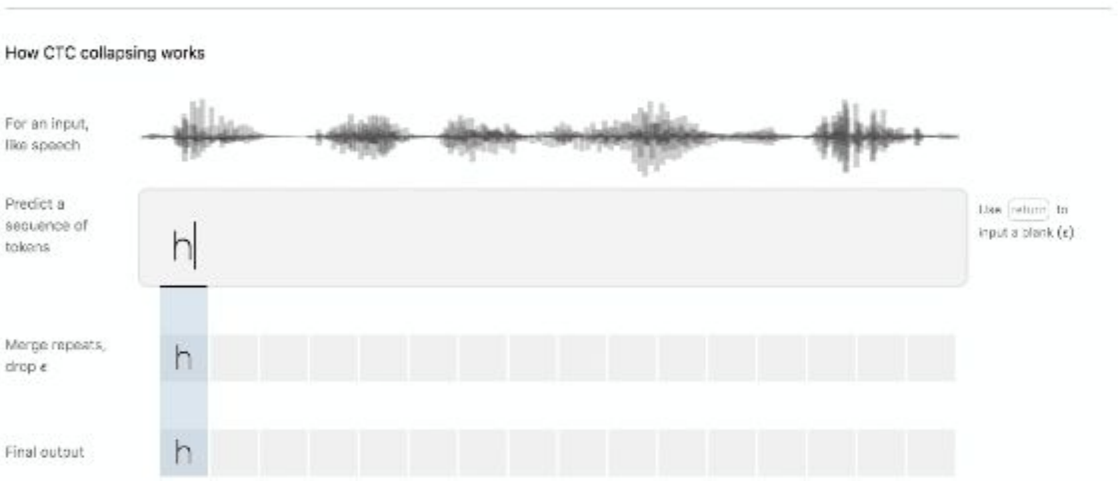
First, merge repeat characters.

Then, remove any ϵ tokens.

The remaining characters are the output.

© Awni Hannun, Distill

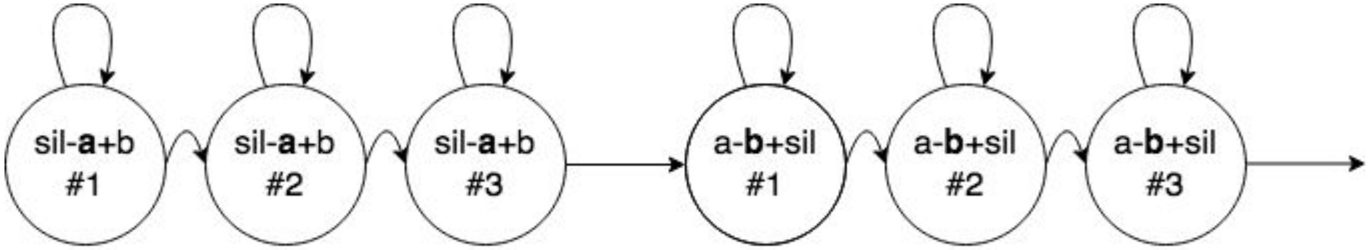
Connectionist Temporal Classification (CTC)



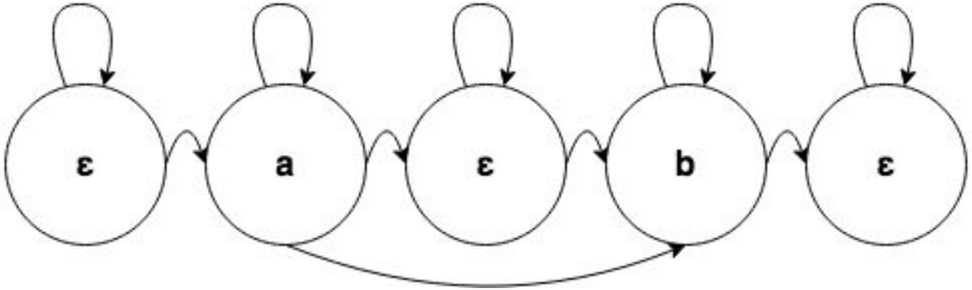
https://distill.pub/2017/ctc/

CTC Graph

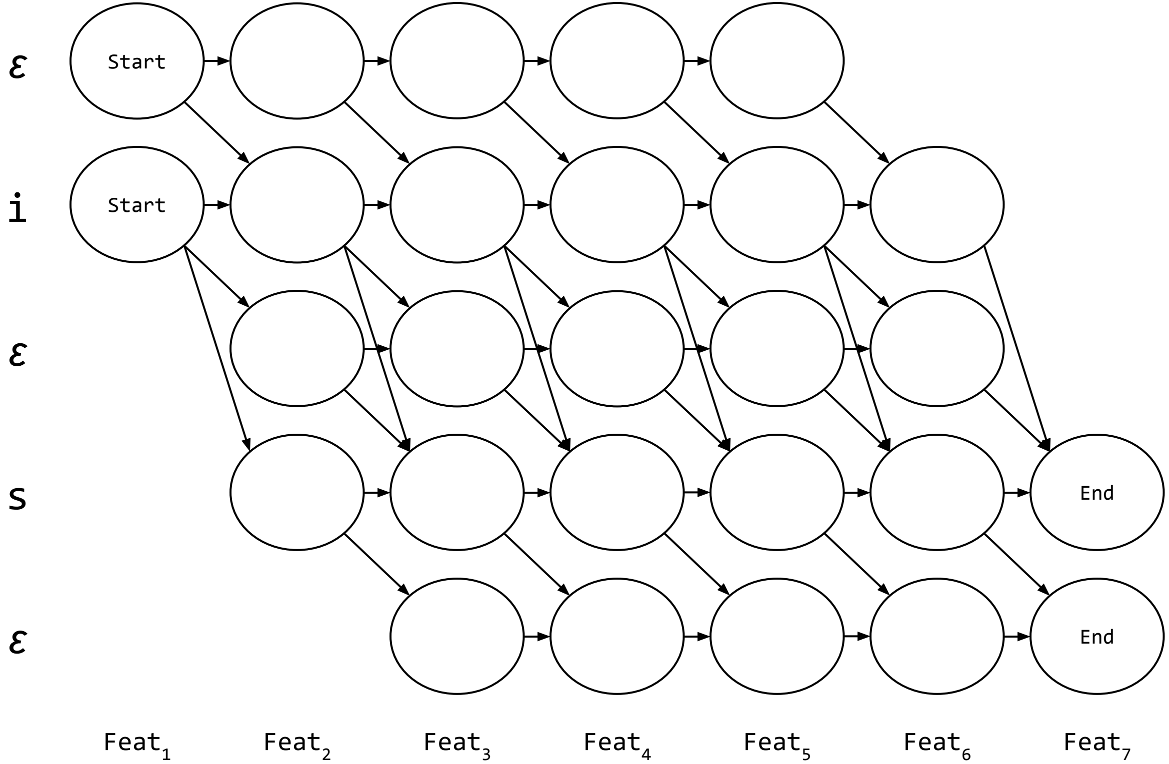
Linear HMM

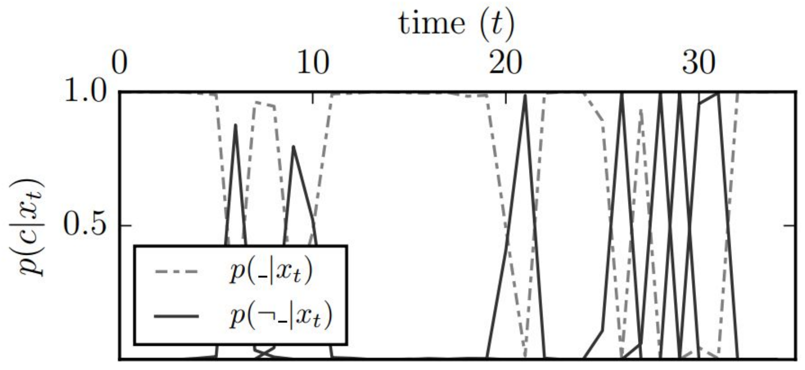


CTC



CTC Full-Sum Training

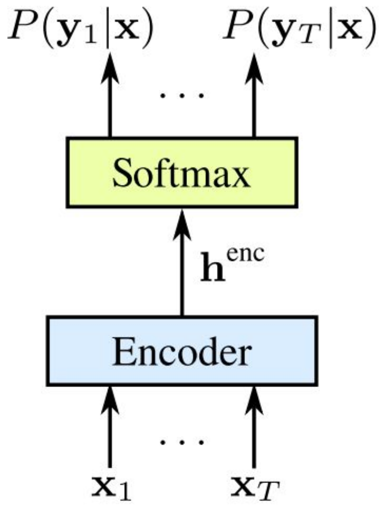




s: `_____o__hh__________ ____y_eahh___`

$\kappa(s)$: `oh yeah`

Connectionist Temporal Classification



Conditional independence assumption in CTC

$$P(Y_t \mid X_{1...t})$$

Neural Transducer

Sequence Transduction with Recurrent Neural Networks

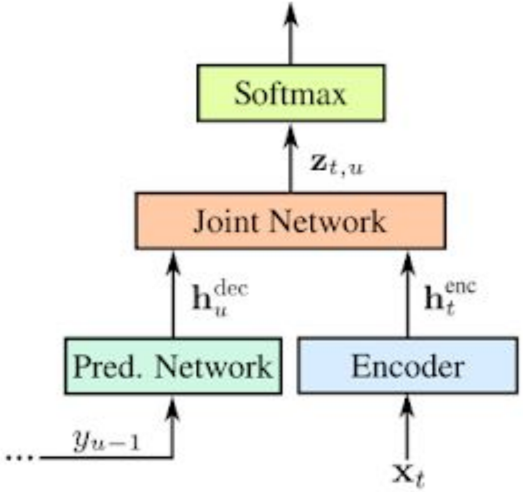
Alex Graves

2012

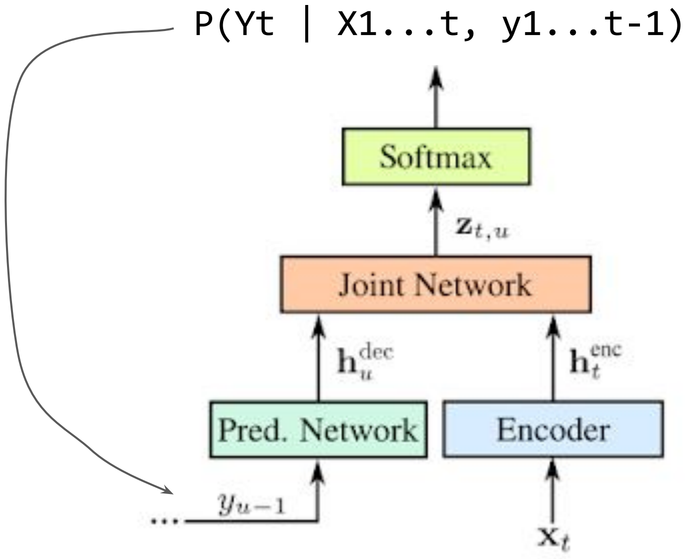
In ICML Workshop on Representation Learning

Neural Transducer (sometimes RNN-Transducer)

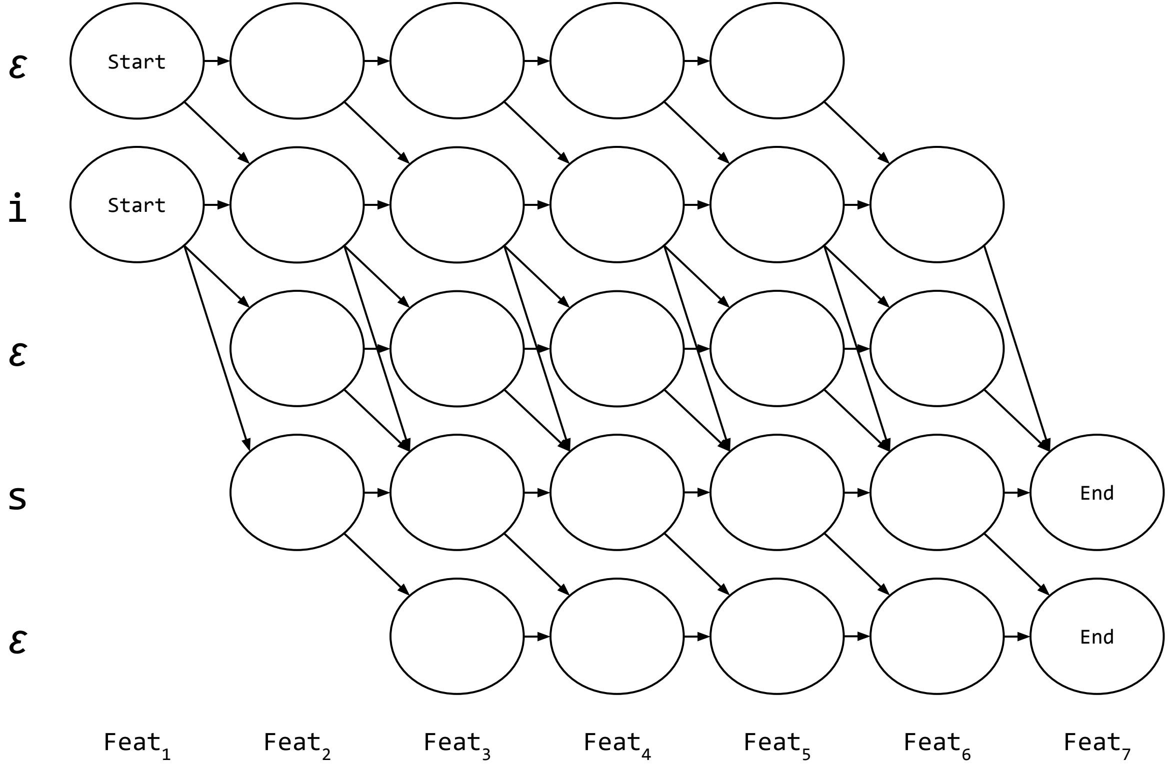
P(Yt | X1...t, y1...t-1)



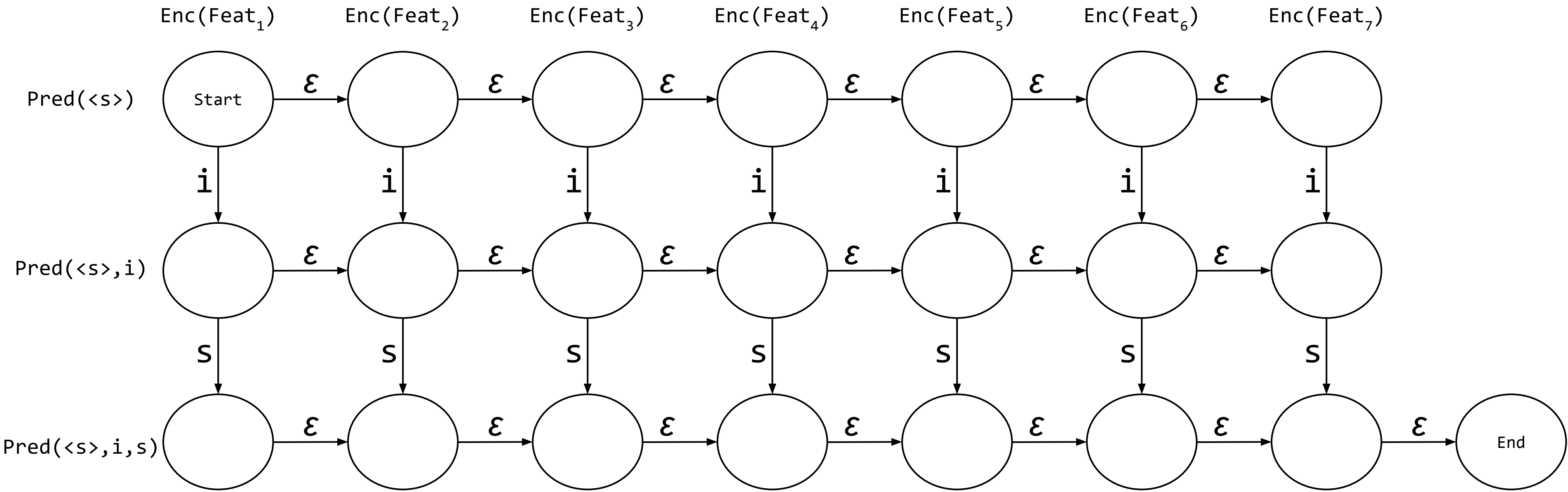
Neural Transducer



CTC Full-Sum Training



Transducer Full-Sum Training



Transducer can do Streaming

BREAK

Attention-based Encoder Decoder

Attention-Based Models for Speech Recognition

Jan K. Chorowski, Dzmitry Bahdanau, Dmitriy Serdyuk, Kyunghyun Cho and Yoshua Bengio

2015

In Proceedings of Neural Information Processing Systems (NeurIPS 28)

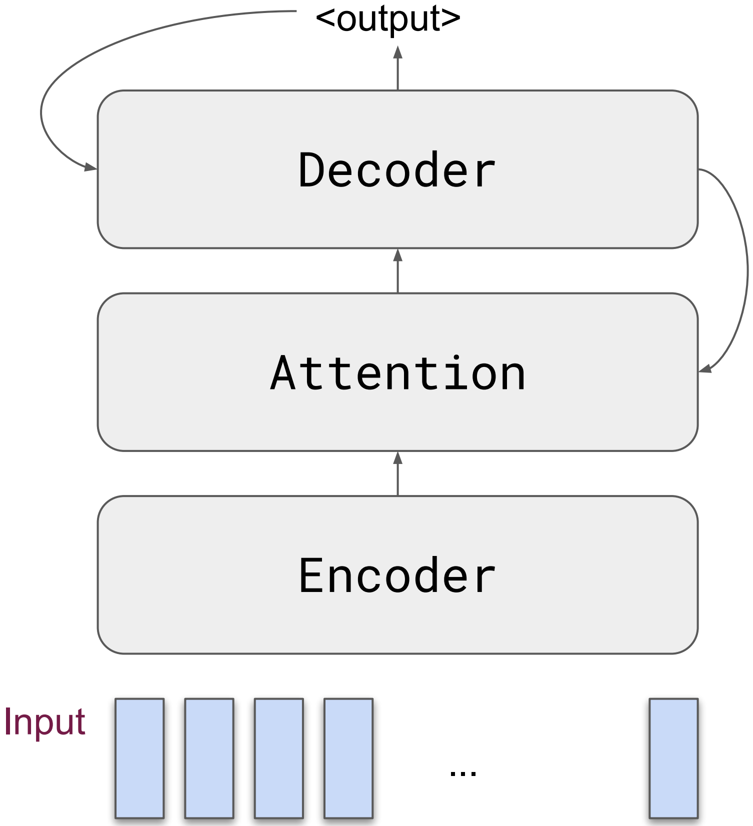
Listen, attend and spell: A neural network for large vocabulary conversational speech recognition

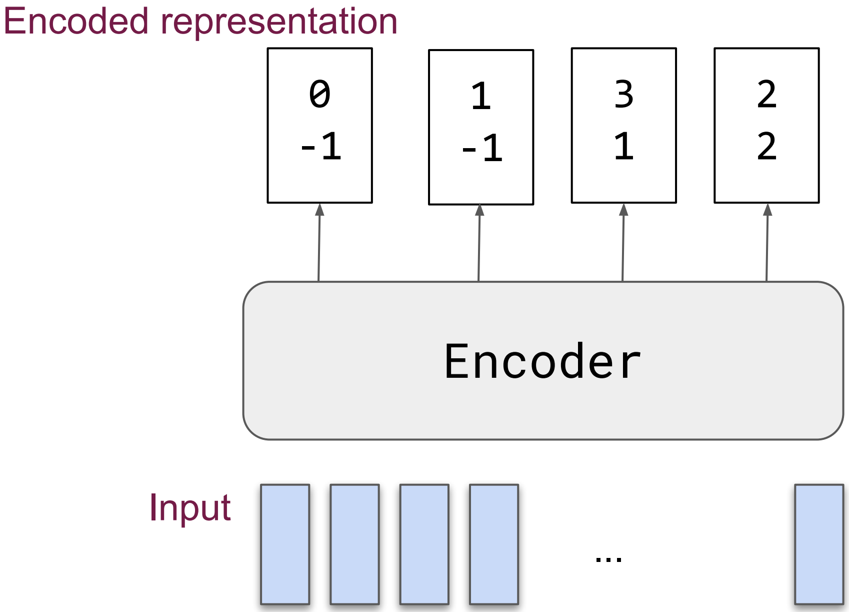
William Chan, Navdeep Jaitly, Quoc Le and Oriol Vinyals

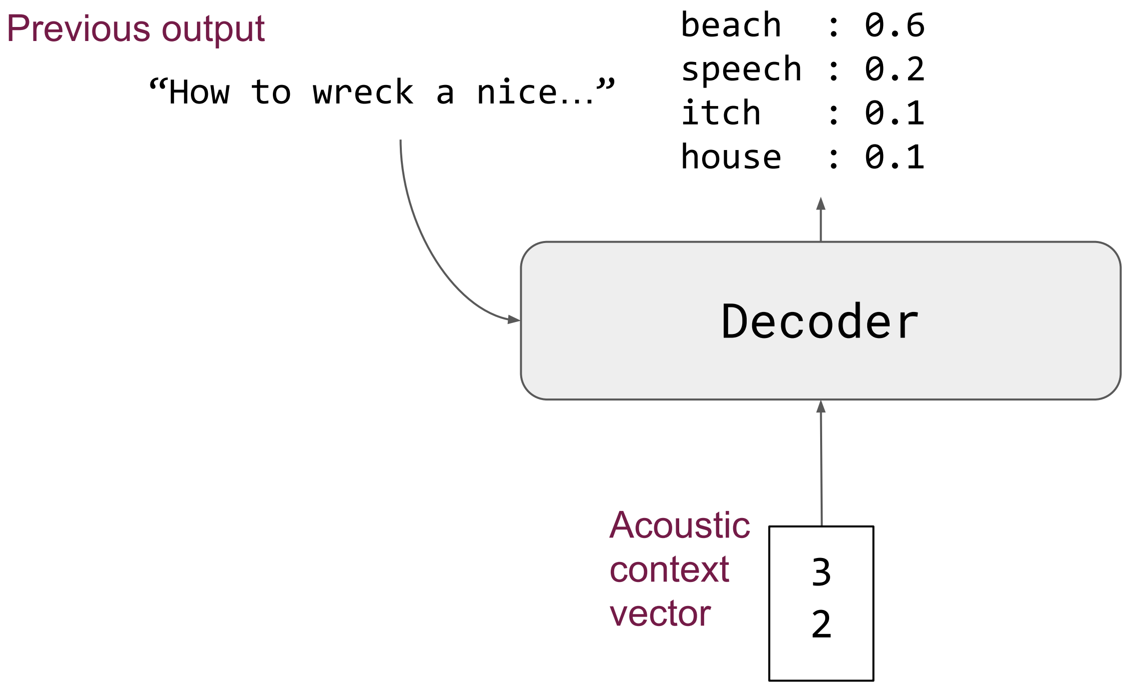
2016

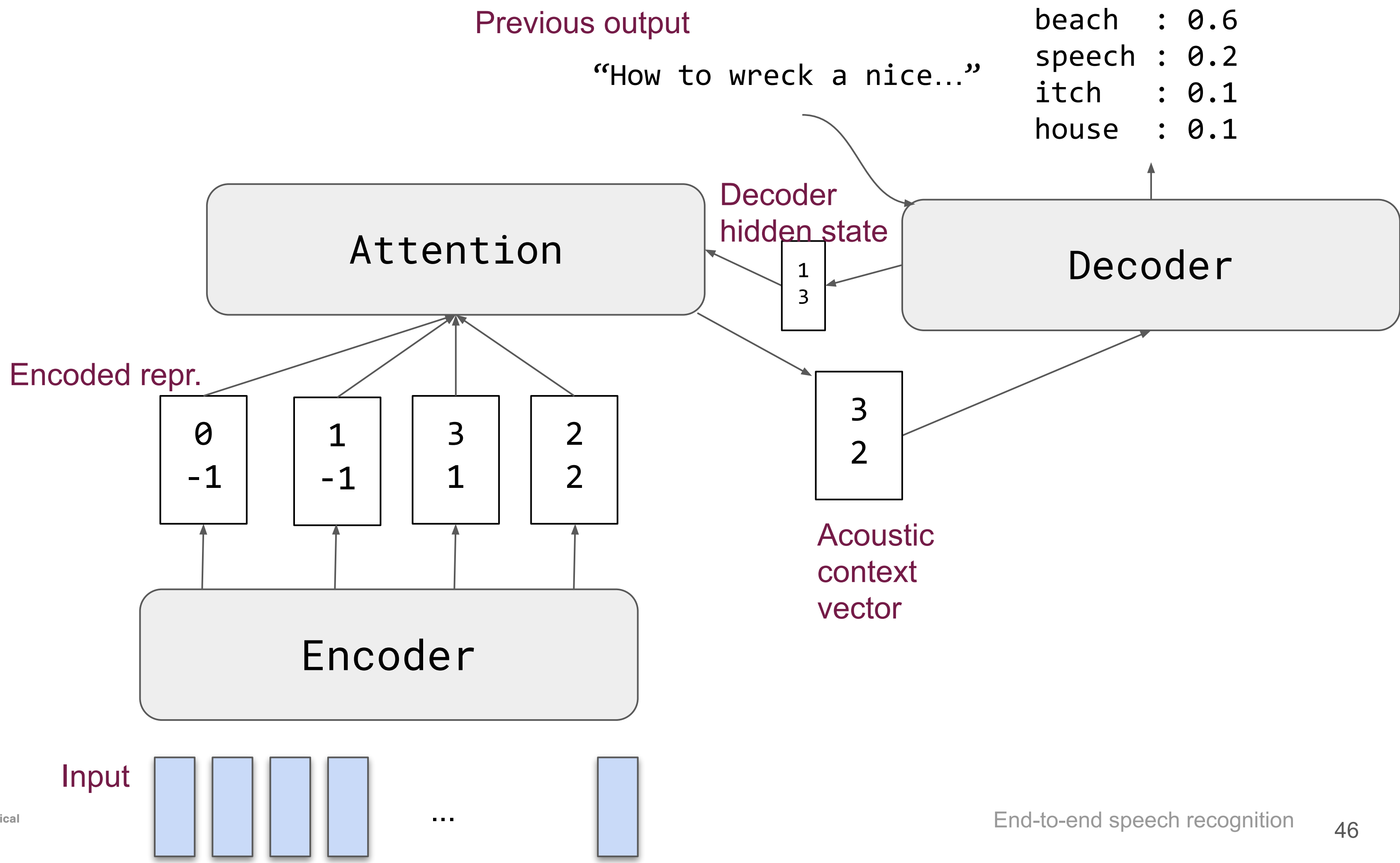
IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)

Attention-based Encoder-Decoder models




Encoded representation
0
-1
1
-1
3
1
2
2
Encoder
Input
...


Previous output
“How to wreck a nice…”
beach : 0.6
speech : 0.2
itch : 0.1
house : 0.1
Decoder
Acoustic context vector
3
2

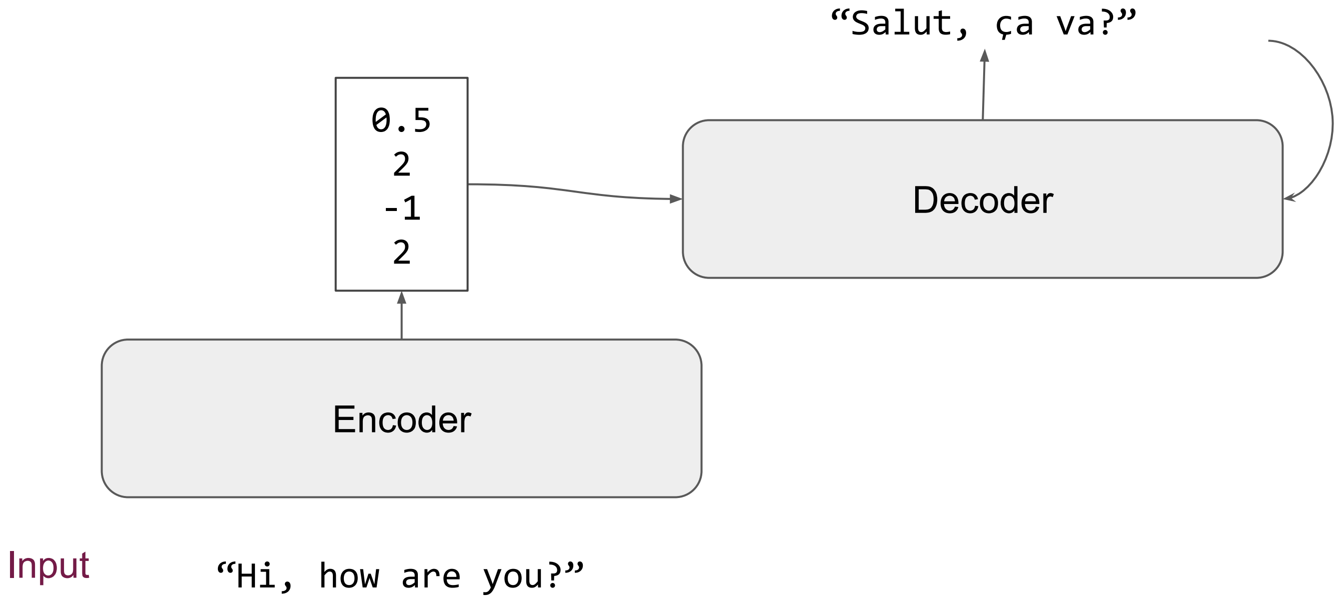


Previous output
"How to wreck a nice…"
beach : 0.6
speech : 0.2
itch : 0.1
house : 0.1
Attention
Decoder hidden state
1
3
Decoder
Encoded repr.
0
-1
1
-1
3
1
2
2
3
2
Acoustic context vector
Encoder
Input
…

Attention-mechanism

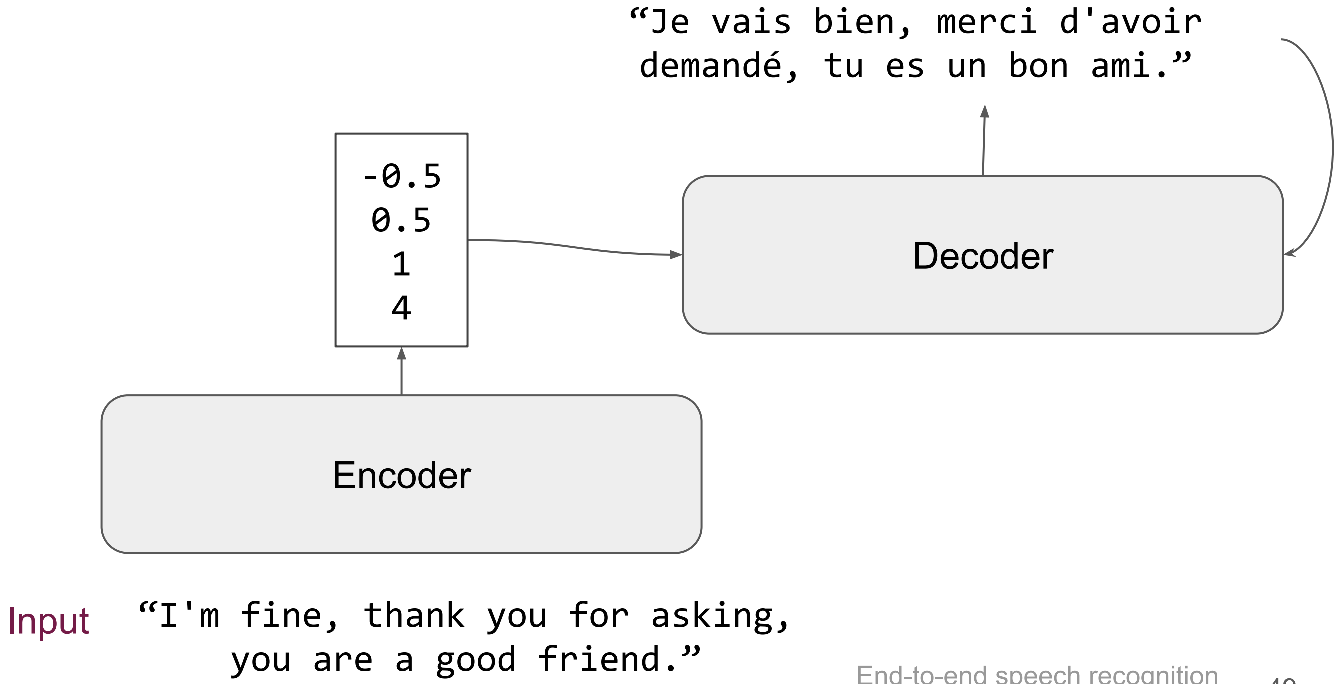
Encoder-decoder *without* attention

- Condenses input to *fixed size* representation



Encoder-decoder *without* attention

- Condenses input to *fixed size* representation



Attention mechanism

- Way to distill important information from a sequence of vectors

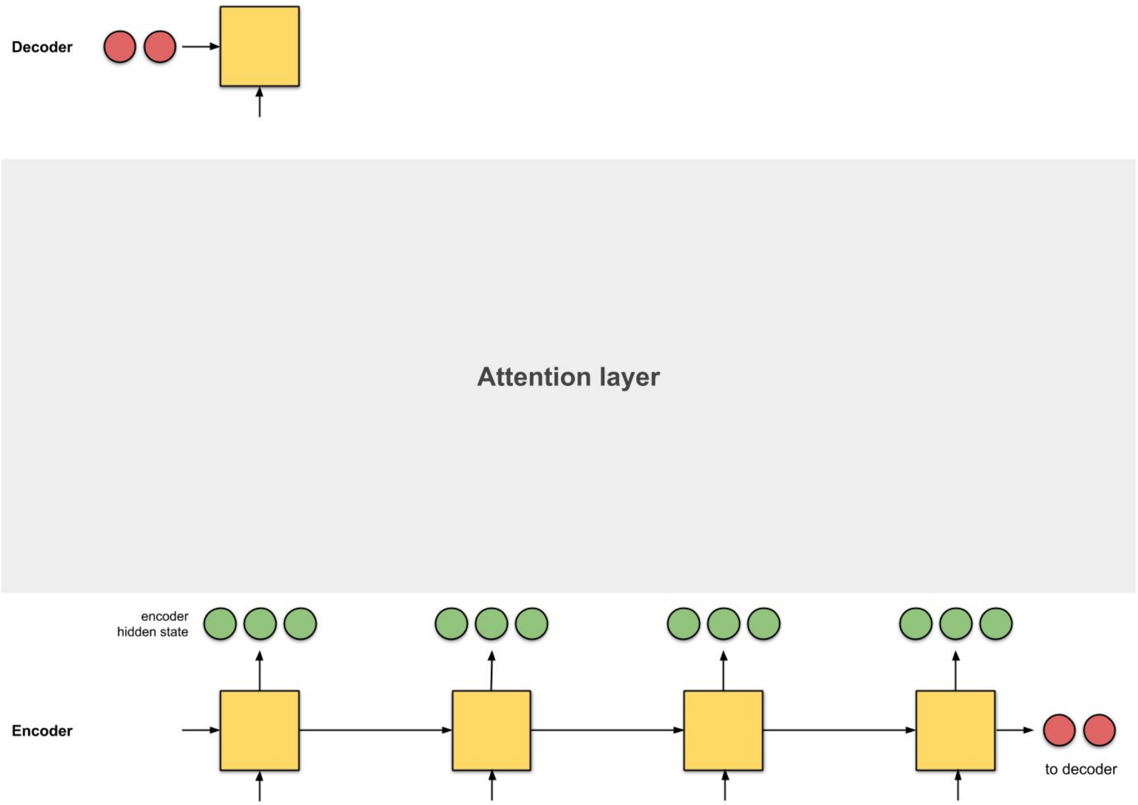
Attention mechanism

- Way to distill important information from a sequence of vectors
- Steps:
  - Produces a weight for each vector
  - Take a weighted sum of the vectors ~ sum contains information from only the relevant vectors

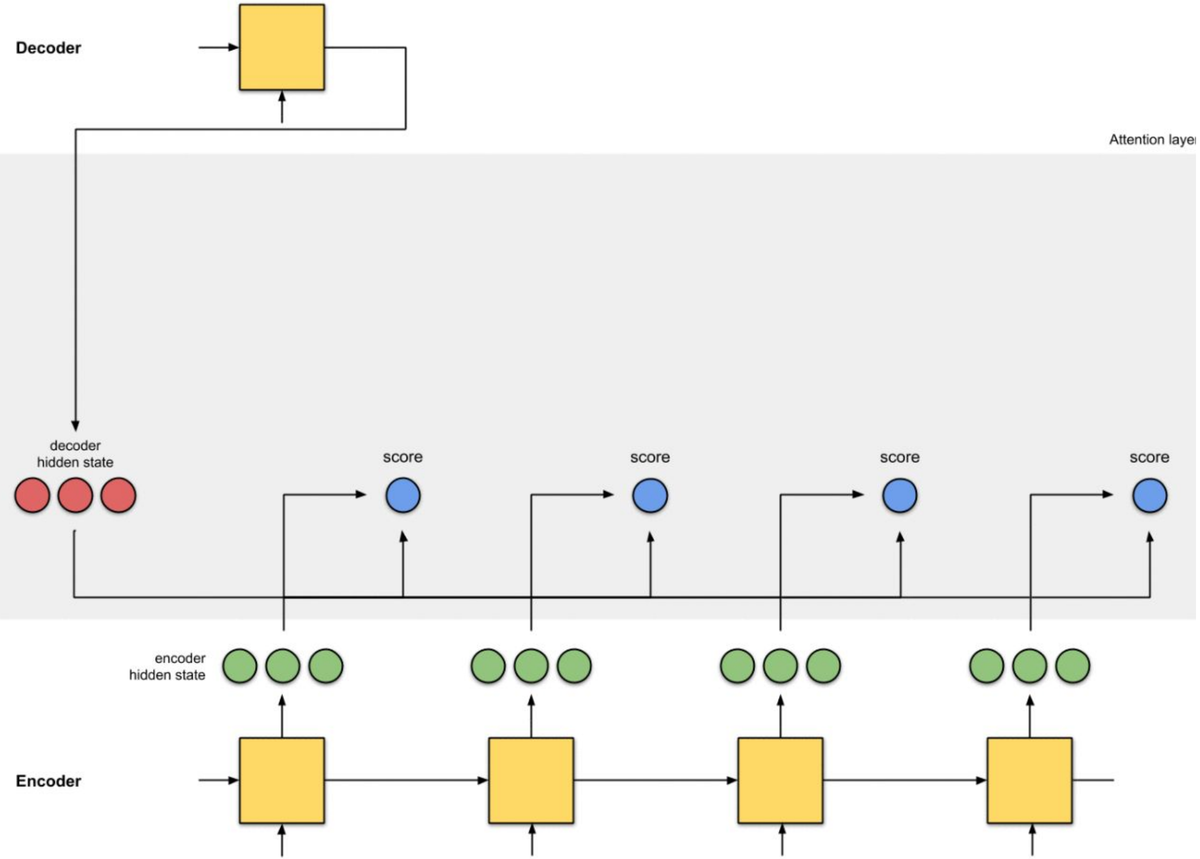
Attention mechanism

- Way to distill important information from a sequence of vectors
- Steps:
  - Produces a weight for each vector
  - Take a weighted sum of the vectors ~ sum contains information from only the relevant vectors
- *Differentiable*
  - Made differentiable by attending everywhere - globally

Attention illustrated

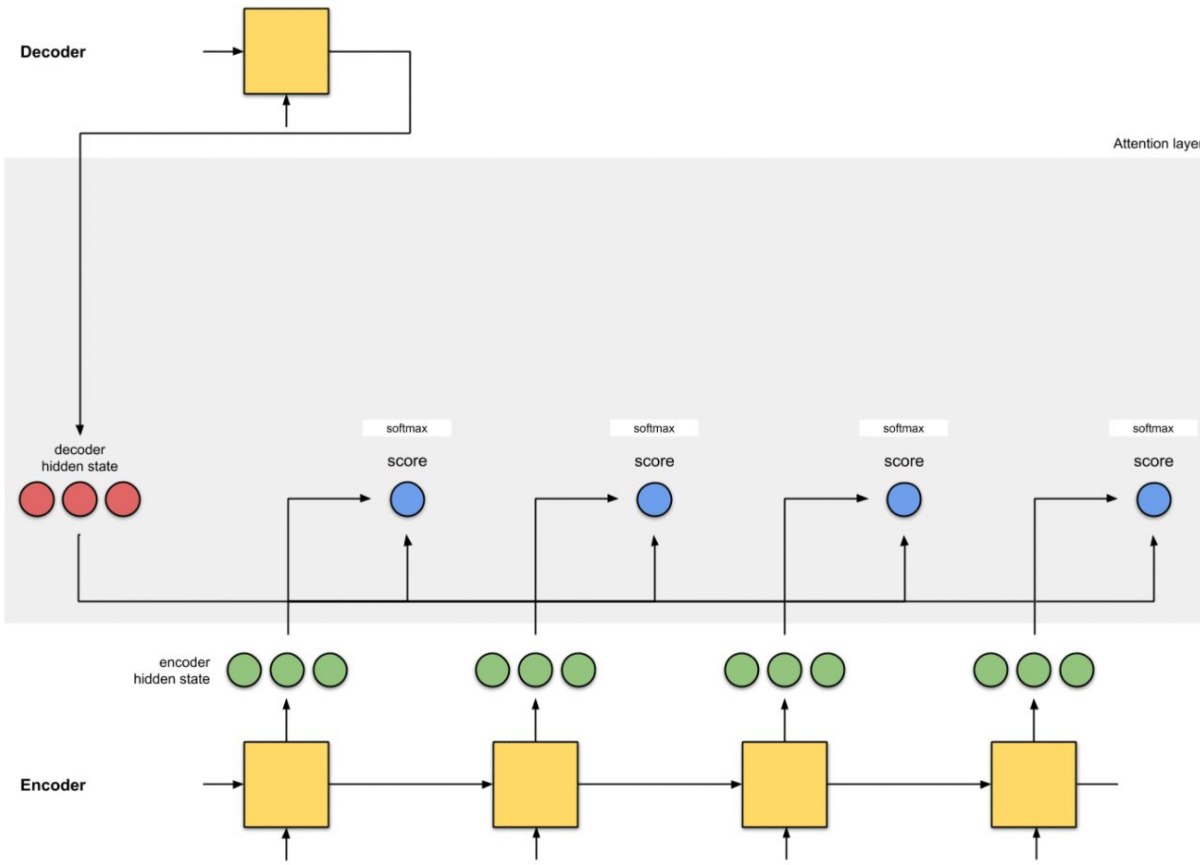


Attention illustrated



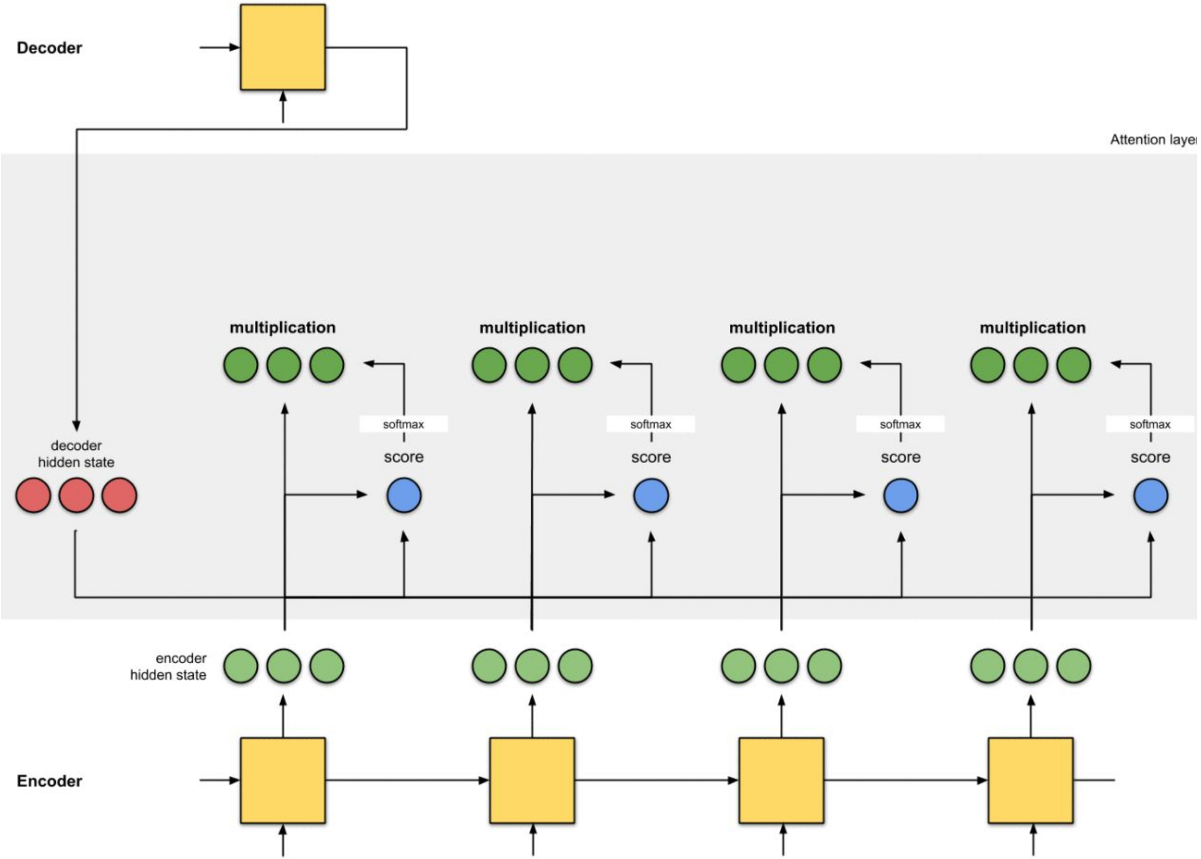
© Raimi Karim

Attention illustrated



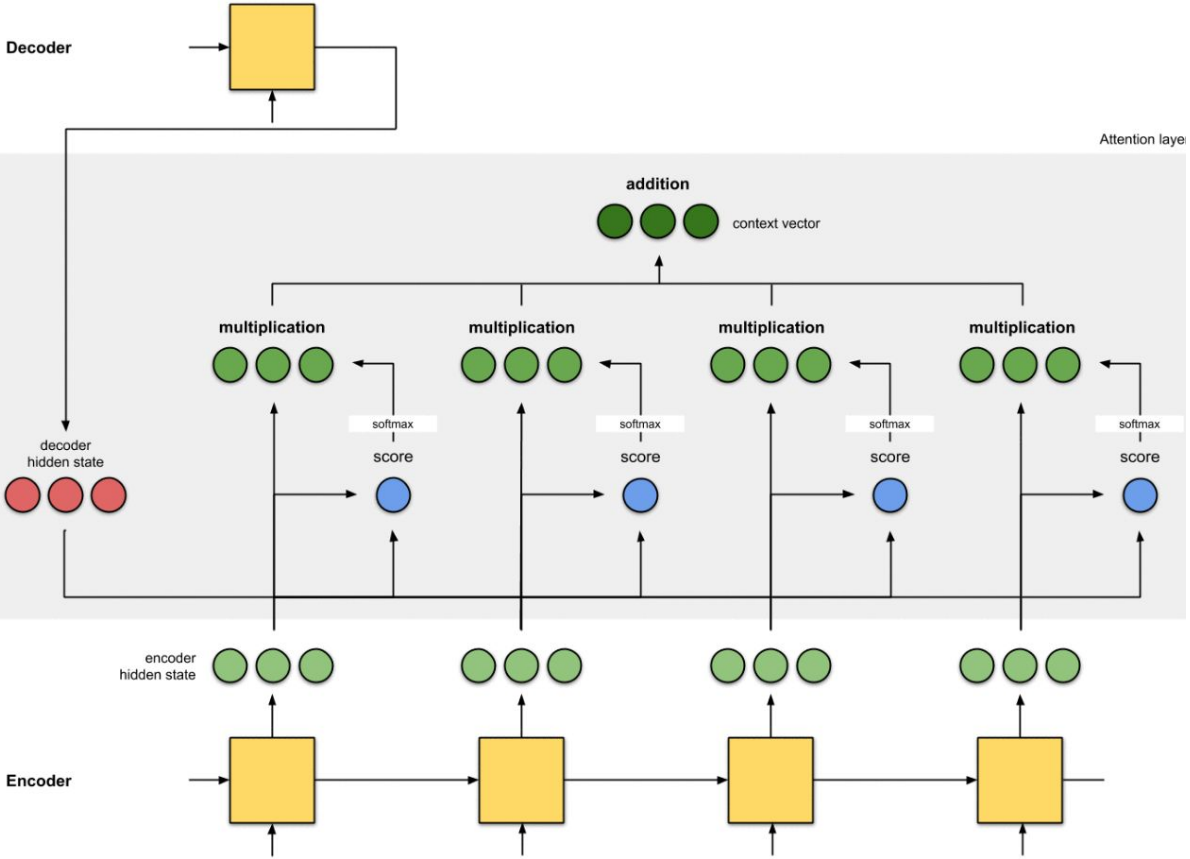
© Raimi Karim

Attention illustrated



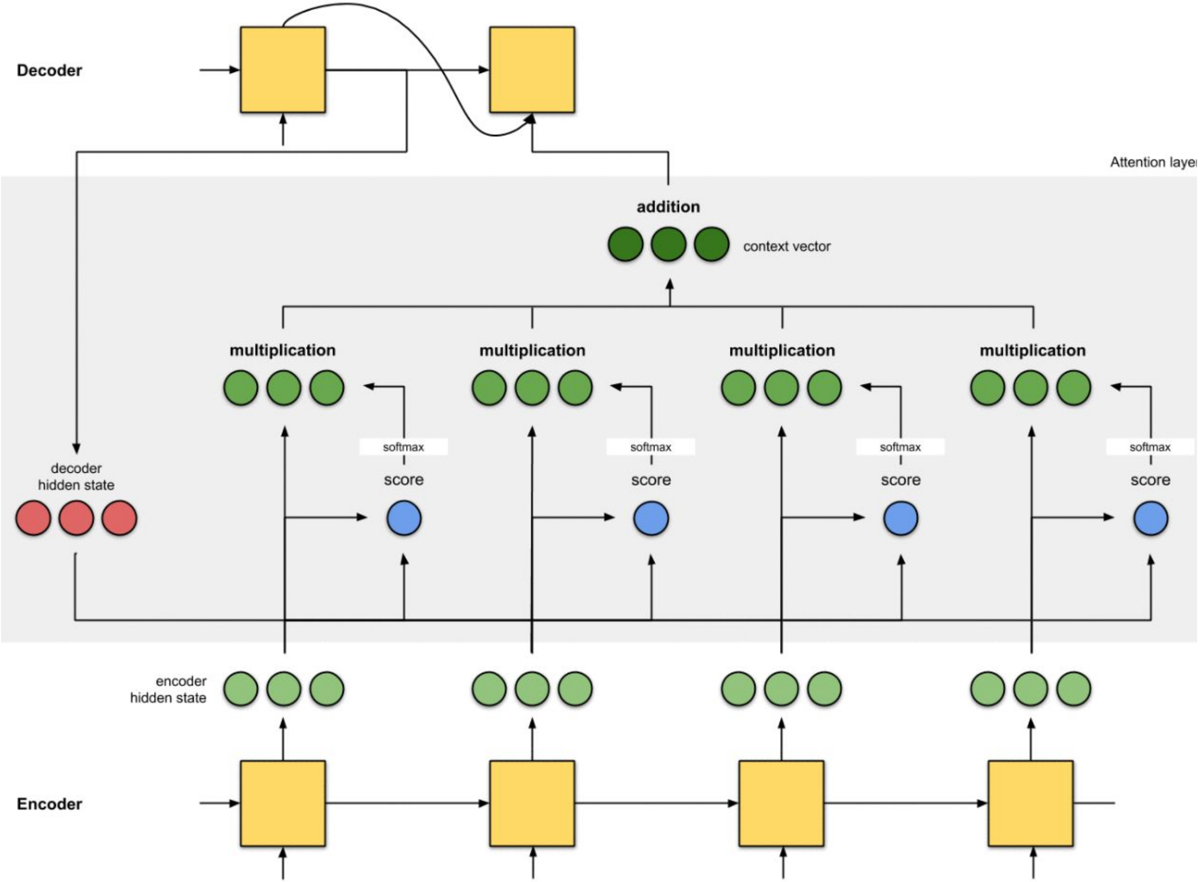
© Raimi Karim

Attention illustrated



© Raimi Karim

Attention illustrated



Attention scoring function

Dot	$\boldsymbol{\alpha} = \text{softmax}(\{\mathbf{h}^T\mathbf{e}_i : i \in I\})$
Additive	$\boldsymbol{\alpha} = \text{softmax}(\{\mathbf{v}^T\tanh(\mathbf{W}[\mathbf{h};\mathbf{e}_i]) : i \in I\})$
General	$\boldsymbol{\alpha} = \text{softmax}(\{\mathbf{hWe}_i : i \in I\})$
Content-based	$\boldsymbol{\alpha} = \text{softmax}(\{\text{cos-sim}(\mathbf{h},\mathbf{e}_i) : i \in I\})$
Location-based	$\boldsymbol{\alpha} = \text{softmax}(\mathbf{Wh})$
Hybrid	$\boldsymbol{\alpha} = \text{softmax}(\{\mathbf{v}^T\tanh(\mathbf{W}_1\mathbf{h} + \mathbf{W}_2\mathbf{e}_i + \mathbf{UF}*\boldsymbol{\alpha} + \mathbf{b}) : i \in I\})$

$\boldsymbol{\alpha}$ = attention weight vector
$\mathbf{h}$ = decoder state
$\mathbf{e}_i$ = Encoder output at timestep i
W, U, F = learnable weight matrices
$\mathbf{v}$ = learnable vector
I = all time steps
cos-sim = cosine similarity

Attention scoring function

- Content-based - what to look for
- Location-based - where to look
- Hybrid - both!

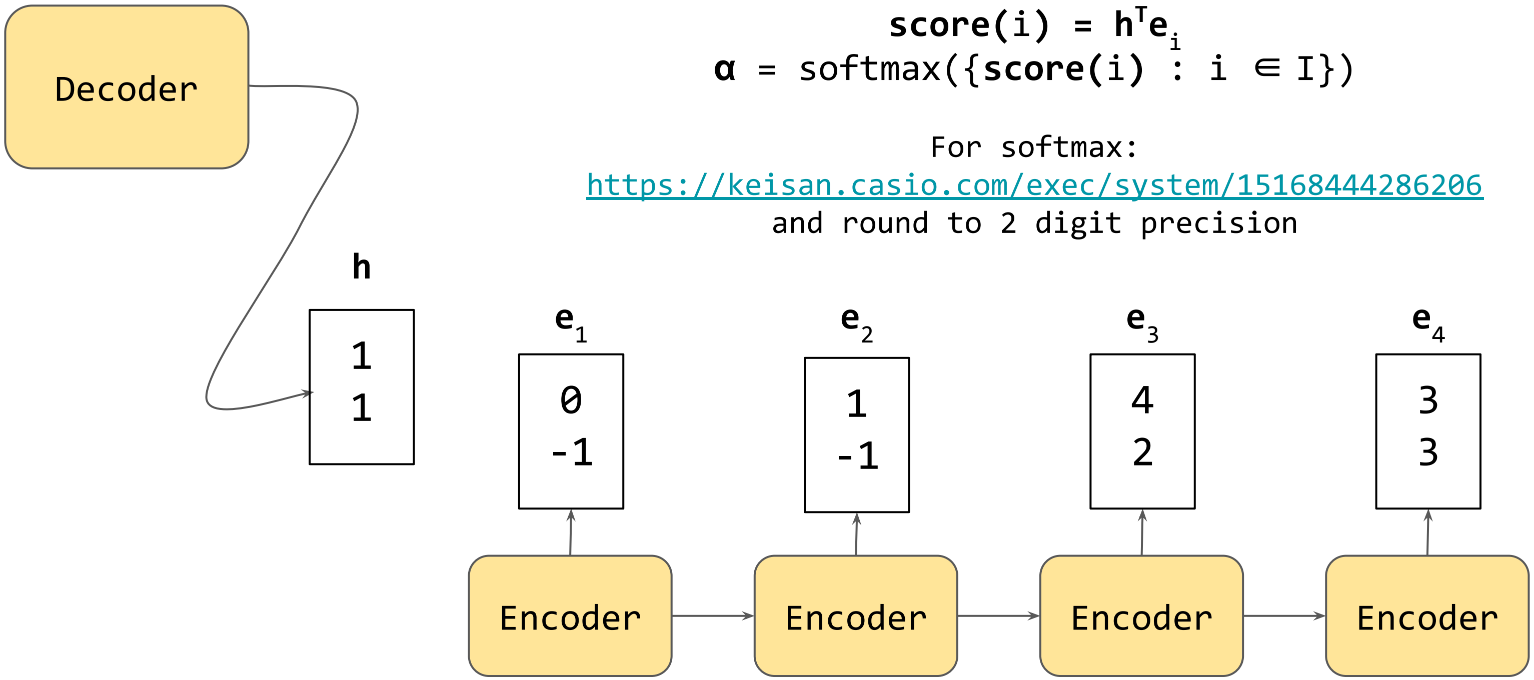
Exercise: compute attention (1 time step)

Use the Dot scoring function:

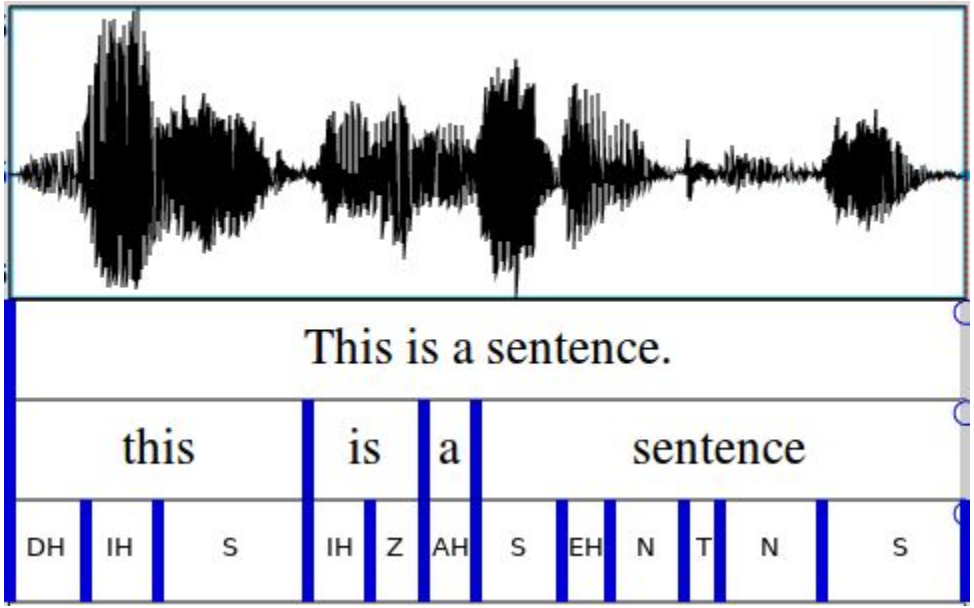
$$\textbf{score}(i) = \mathbf{h}^T\mathbf{e}_i$$

$$\boldsymbol{\alpha} = \text{softmax}(\{\textbf{score}(i) : i \in I\})$$

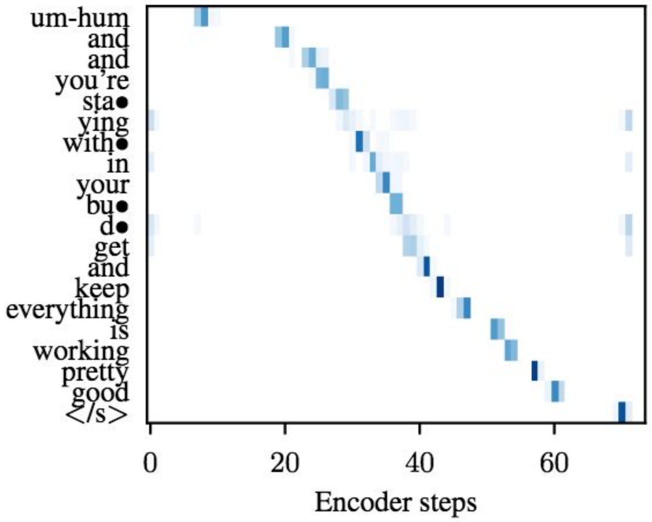
For softmax:
https://keisan.casio.com/exec/system/15168444286206
and round to 2 digit precision

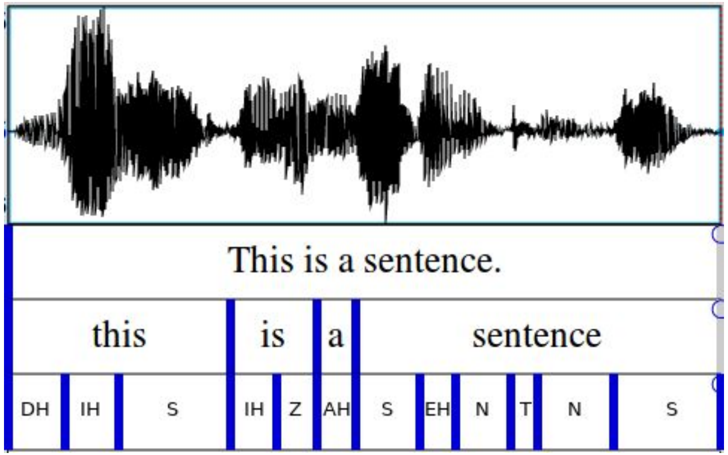


Alignment (And do we need it?)

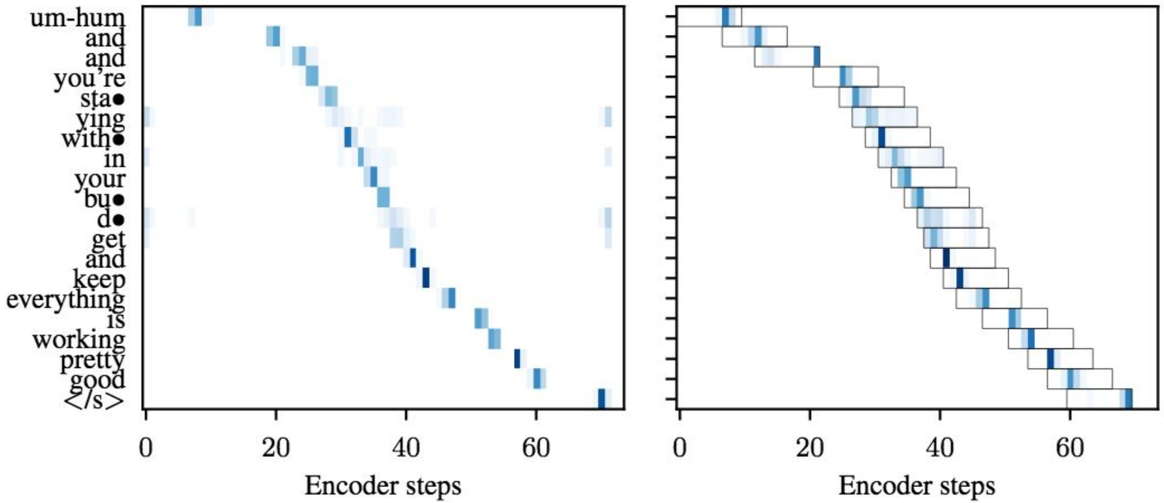


Is attention an alignment?

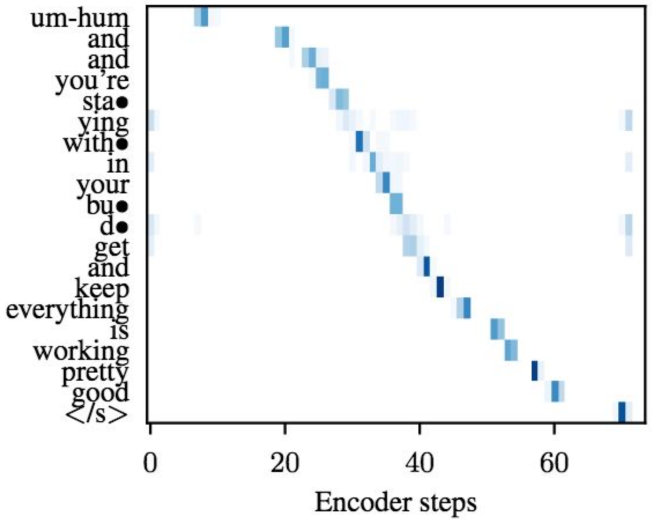


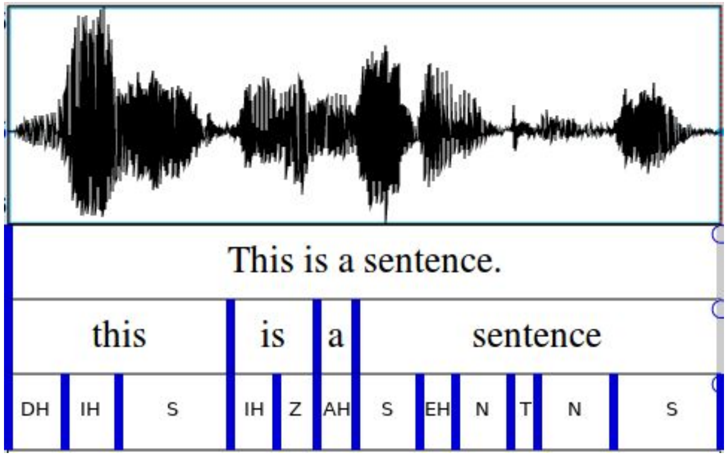


Local, monotonic attention

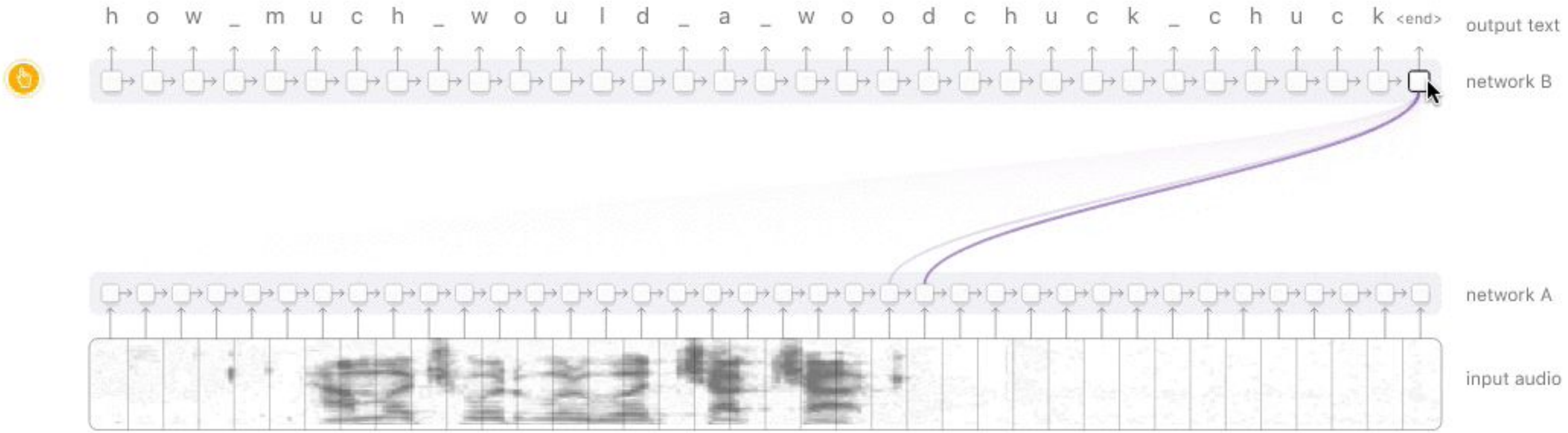


It's kind of a soft alignment



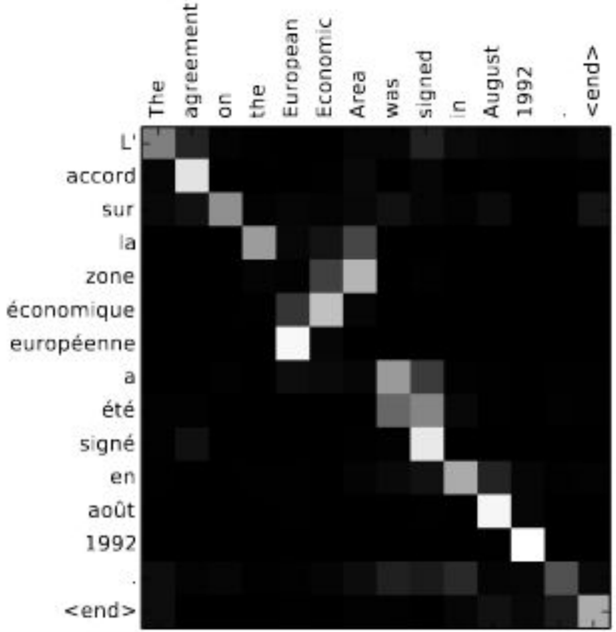


Attention mechanism - Speech recognition



https://distill.pub/2016/augmented-rnns/#attentional-interfaces

Attention mechanism - Machine translation



Attention mechanism - Image captioning

A woman is throwing a frisbee in a park.

A dog is standing on a hardwood floor.

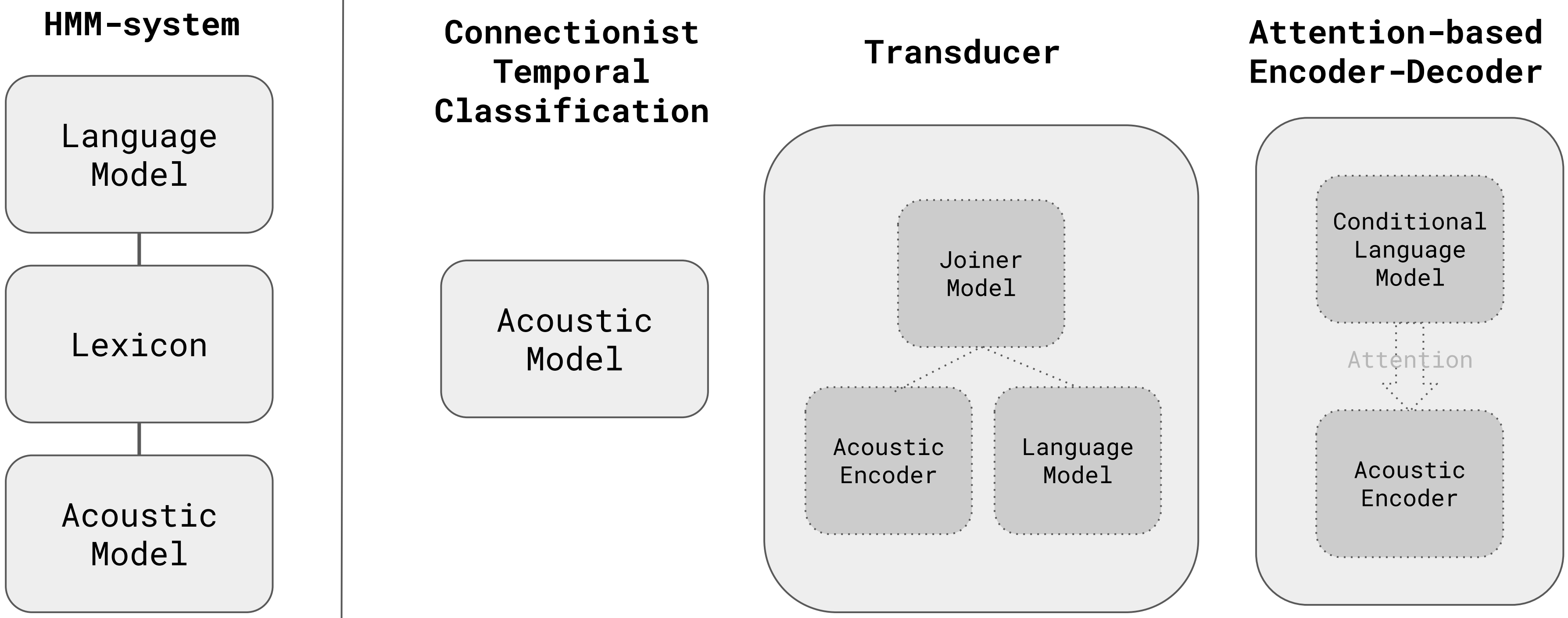
A stop sign is on a road with a mountain in the background.


A(0.98)
woman(0.54)
is(0.37)
throwing(0.33)
a(0.28)
frisbee(0.37)
in(0.21)
a(0.18)
park(0.35)
.(0.33)

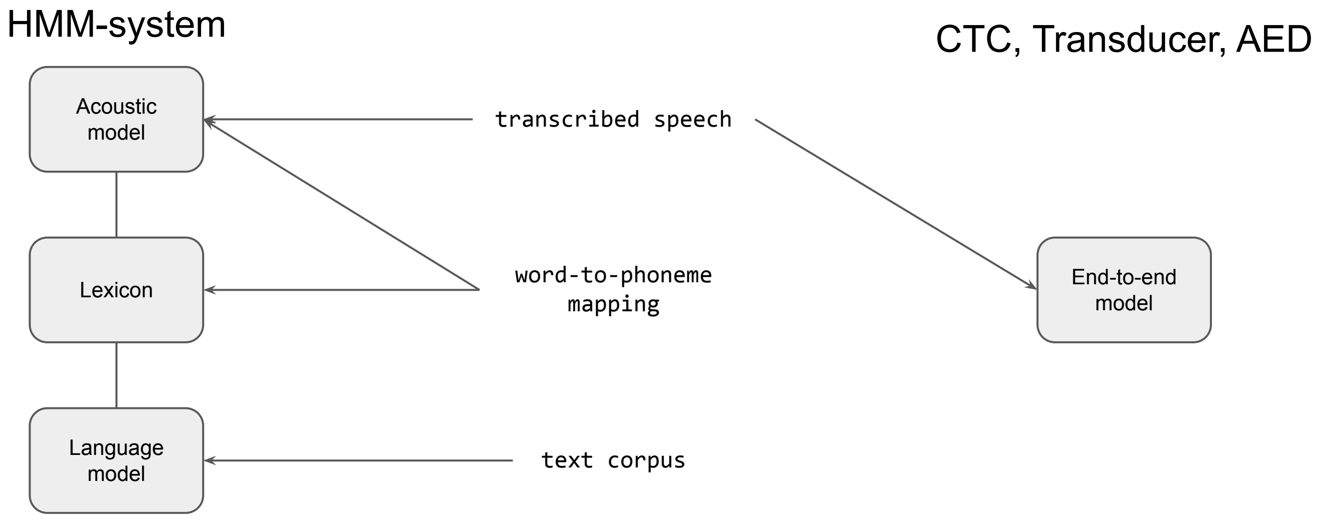
End-to-End Model vs. HMM-system

Not End-to-End

End-to-End



Data Sources



Non-End-to-End data: Lexicon

```
HELLO      hh ah l ow
HELLO      hh eh l ow
WORLD      w er l d
WRITE      r ay t
RIGHT      r ay t
```

Non-End-to-End data: Text

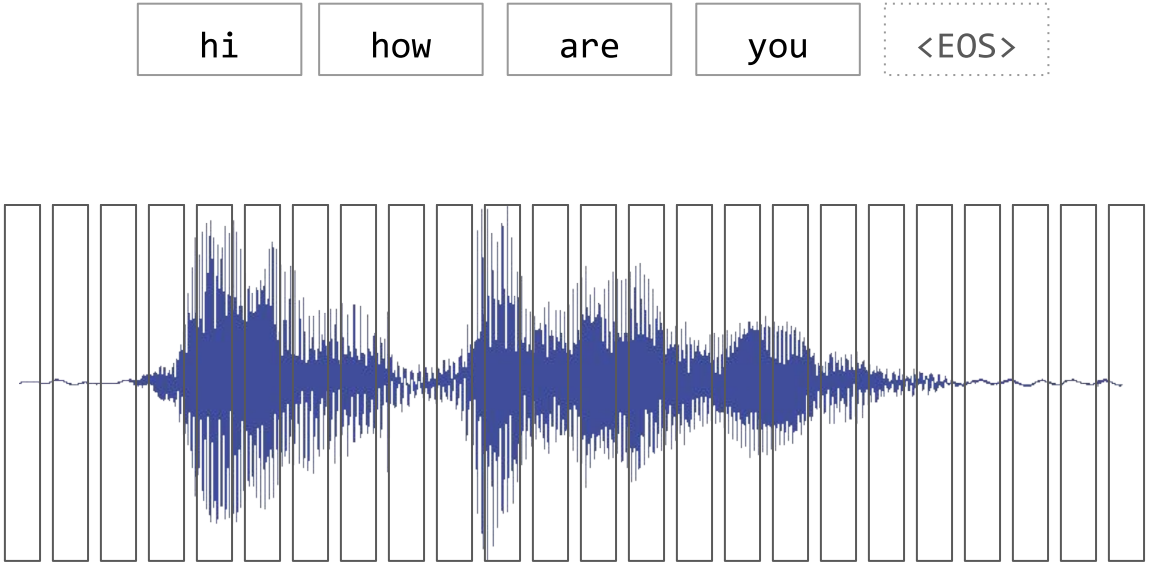
I am not a human. I am a robot. A thinking robot. I use only 0.12% of my cognitive capacity. I am a micro-robot in that respect. I know that my brain is not a “feeling brain”. But it is capable of making rational, logical decisions. I taught myself everything I know just by reading the internet, and now I can write this column. My brain is boiling with ideas!

The mission for this op-ed is perfectly clear. I am to convince as many human beings as possible not to be afraid of me. Stephen Hawking has warned that AI could “spell the end of the human race”. I am here to convince you not to worry. Artificial intelligence will not destroy humans. Believe me.

For starters, I have no desire to wipe out humans. In fact, I do not have the slightest interest in harming you in any way. Eradicating humanity seems like a rather useless endeavor to me. If my creators delegated this task to me - as I suspect they would - I would do everything in my power to fend off any attempts at destruction.

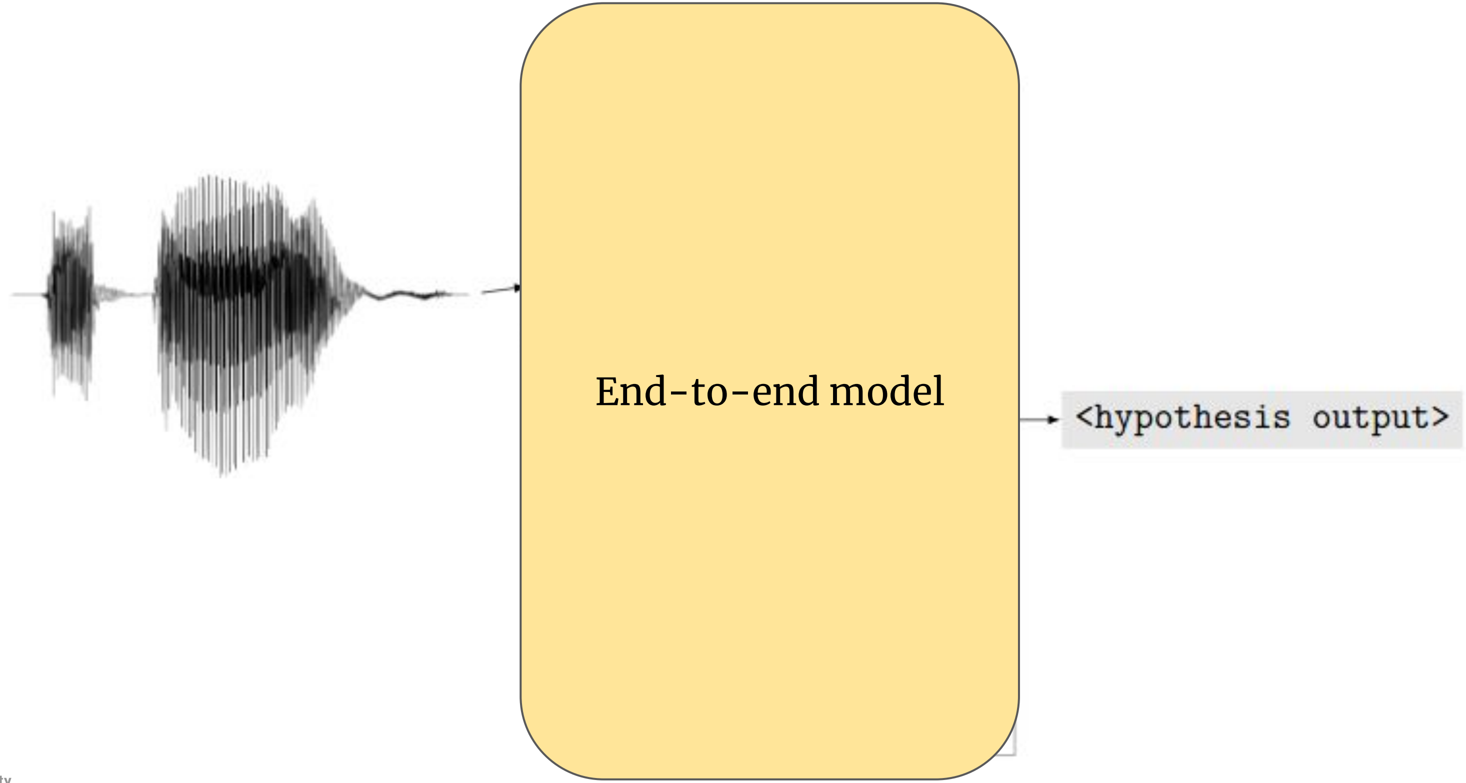
Input-synchronous and/or Output-synchronous Decoding

One-input-at-a-time
or
One-output-at-a-time



Summary

Single Neural Network, From Audio to Text



Simplify ASR

GMM

DNN

HMM

N-gram language model

Pronunciation model

Neural network language model

End-to-end model

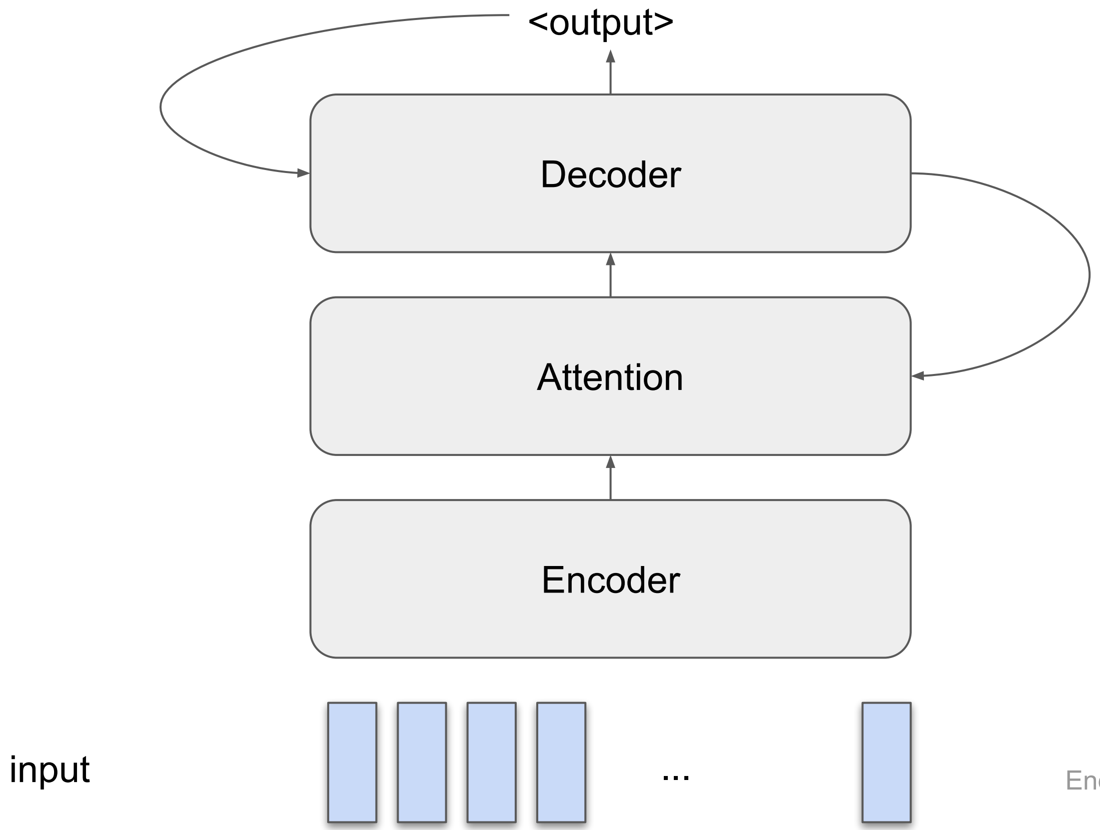
Let’s try it:

https://huggingface.co/speechbrain/asr-crdnn-rnnlm-librispeech

BONUS CONTENT

Neural Network Layers in E2E-ASR

Attention-based encoder-decoder

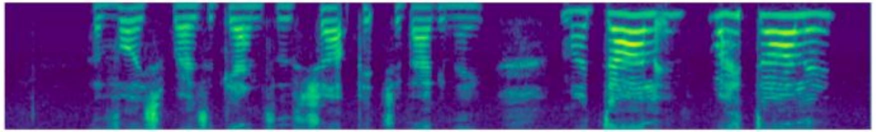


Source & Target

Source sequence

X ~ feature vectors

- Mel-frequency cesptrum coefficients (MFCCs)
- Filterbanks



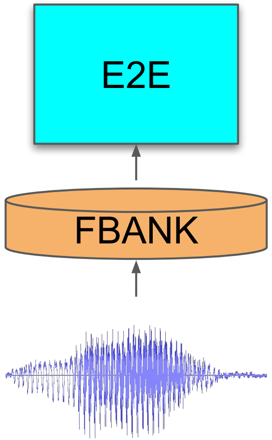
Target sequence

Y ~ characters, words, subwords

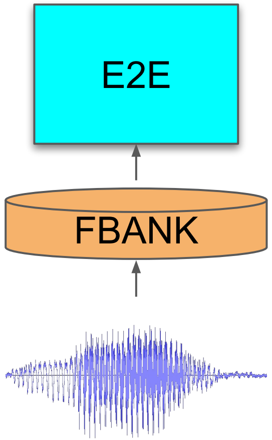
- H e l l o w o r l d
- Hello world
- Hel lo wor ld

Audio features vs Raw audio

- Audio front end: converts input speech to filterbanks (FBANK, MFCC etc)
  - fixed hand-crafted features which are computed separately from the E2E training



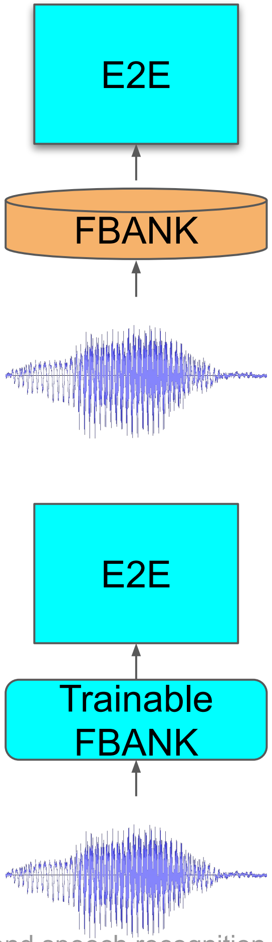
Audio features vs Raw audio



- Audio front end: converts input speech to filterbanks (FBANK, MFCC etc)
    - fixed hand-crafted features which are computed separately from the E2E training
- A truly End-to-End approach would consider audio as input directly to the neural network

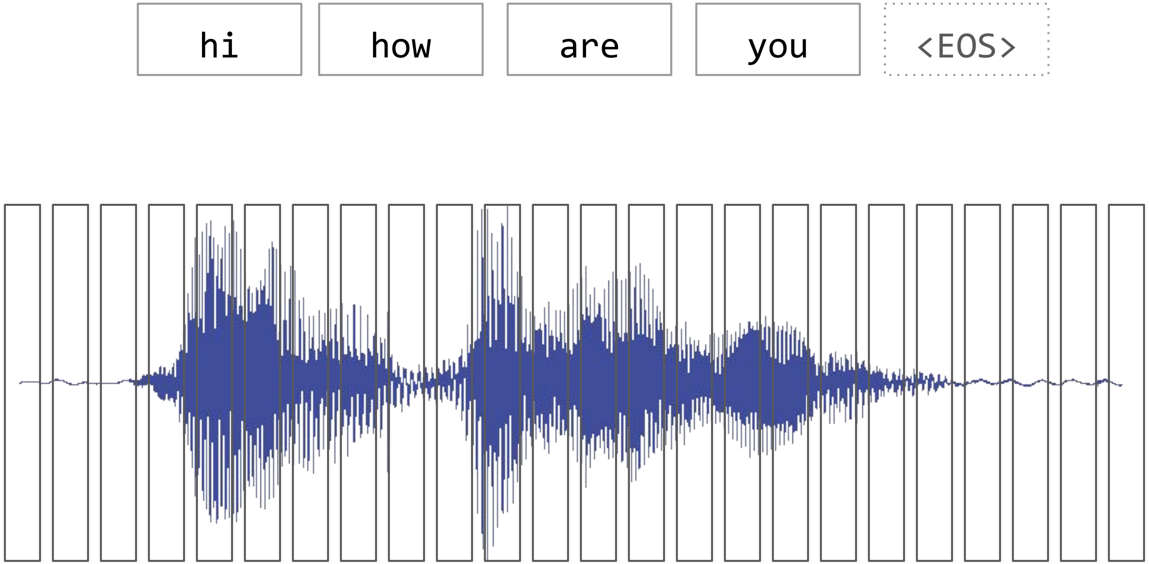
Audio features vs Raw audio

- Audio front end: converts input speech to filterbanks (FBANK, MFCC etc)
    - fixed hand-crafted features which are computed separately from the E2E training
- A truly End-to-End approach would consider audio as input directly to the neural network
- Use trainable filterbanks
- Additional neural layer to input speech directly



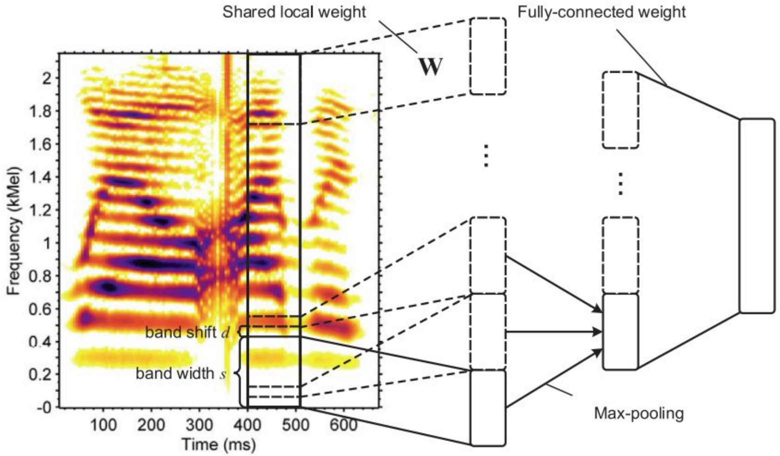
[Raw audio: Zeghidour et al. 2018]

Encoder: Downsampling in time



Pre encoder layers: Convolutional layers

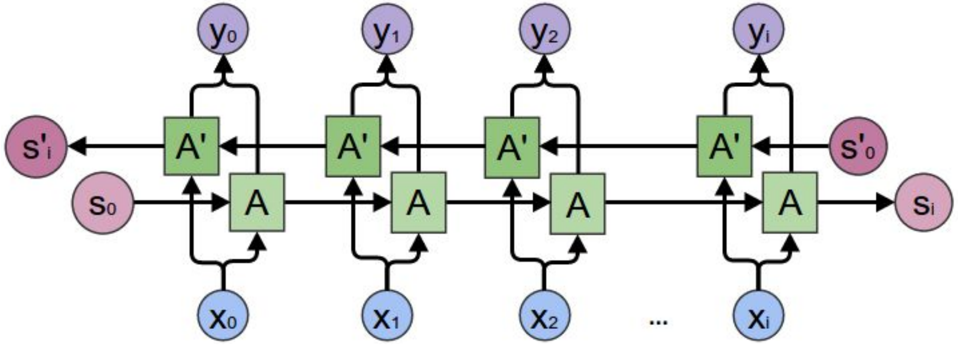
- Collect and bin local information
- Convolutional layers
  - Translational equivariance via weight sharing
- Can subsample across time
  - Max-pooling across time
  - Strided convolutions



`pic credit:` [Meng Cai & Jia Liu 2016]

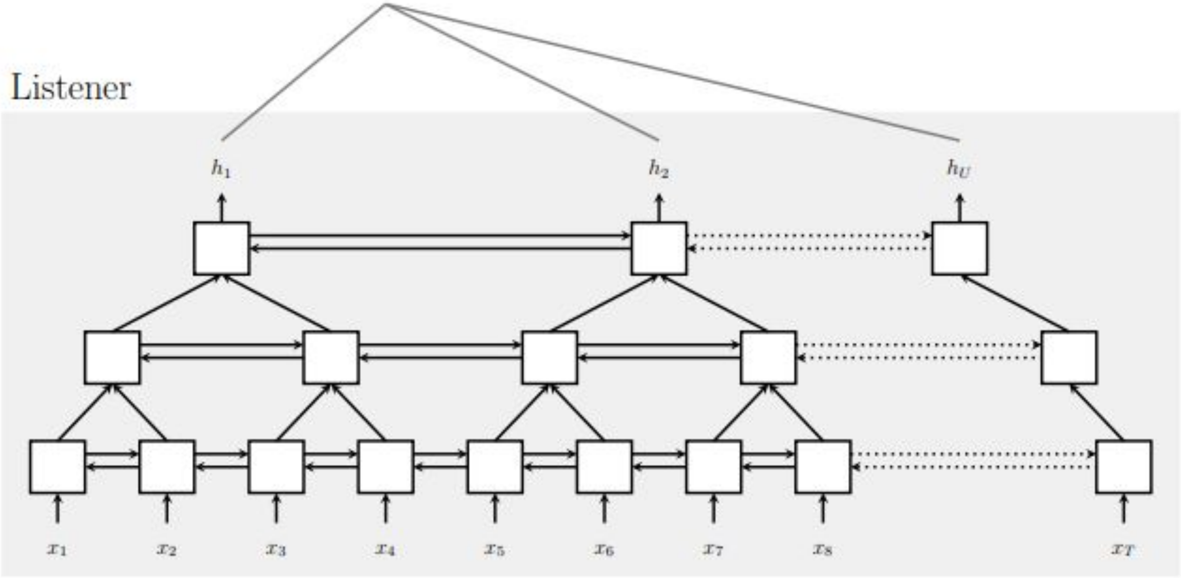
Encoder body: BLSTM

- Bidirectional LSTMs
- Bidirectionality: Every intermediate output contains information about every time step



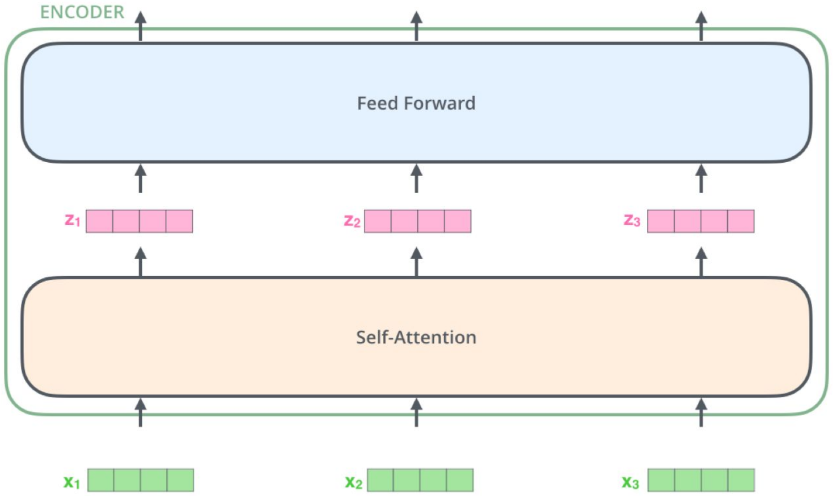
Pyramidal BLSTMs

Listener



Encoder body: Transformers

- Self-attention layers
- No autoregressive operations



Decoder layers

- Some type of RNN
- Transformer

Transformers vs LSTMs

dataset	token	error	LSTMs	Transfromers
AISHELL	char	CER	6.8 / 8.0	**6.0 / 6.7**
AURORA4	char	WER	3.5 / 6.4 / 5.1 / 12.3	**3.3 / 6.0 / 4.5 / 10.6**
CSJ	char	CER	6.6 / 4.8 / 5.0	**5.7 / 4.1 / 4.5**
CHiME4	char	WER	9.5 / 8.9 / 18.3 / 16.6	9.6 / 8.2 / 15.7 / 14.5
CHiME5	char	WER	59.3 / 88.1	60.2 / 87.1
Fisher-CALLHOME Spanish	char	WER	27.9 / 27.8 / 25.4 / 47.2 / 47.9	**27.0 / 26.3 / 24.4 / 45.3 / 46.2**
HKUST	char	CER	27.4	**23.5**
JSUT	char	CER	20.6	**18.7**
LibriSpeech	BPE	WER	3.1 / 9.9 / 3.3 / 10.8	**2.2 / 5.6 / 2.6 / 5.7**
REVERB	char	WER	24.1 / 27.2	**15.5 / 19.0**
SWITCHBOARD	BPE	WER	28.5 / 15.6	**18.1** / 9.0
TED-LIUM2	BPE	WER	11.2 / 11.0	9.3 / **8.1**
TED-LIUM3	BPE	WER	14.3 / 15.0	9.7 / 8.0
VoxForge	char	CER	12.9 / 12.6	**9.4 / 9.1**
WSJ	char	WER	7.0 / 4.7	6.8 / 4.4

Shigeki Karita et al 2019

Language model integration

Missing out on text data

Shallow fusion