Google

Towards End-to-End Speech Recognition

Tara N. Sainath

July 12, 2019

Acknowledgements

Google Brain Team

- Raziel Alvarez
- William Chan
- Jan Chorowski
- Chung-Cheng Chiu
- Zhifeng Chen
- Navdeep Jaitly
- Anjuli Kannan
- Patrick Nguyen
- Ruoming Pang
- Colin Raffel
- Ron Weiss
- Yonghui Wu
- Yu Zhang

Google Speech Team

- Michiel Bacchiani
- Tom Bagby
- Deepti Bhatia
- Alexander Gruenstein
- Shankar Kumar
- Qiao Liang
- Ian McGraw
- Golan Pundak
- Rohit Prabhavalkar
- Kanishka Rao
- David Rybach
- Tara Sainath
- Johan Schalkwyk
- Vlad Schogol
- (June) Yuan Shangguan
- Khe Chai Sim
- Gabor Simko
- Trevor Strohman
- Zelin Wu
- Ding Zhao
- Kazuki Irie (intern)
- Shubham Toshniwal (intern)

Acknowledgement

The content of this tutorial is mostly based on the following tutorial with recent updates.

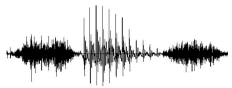
End-to-End Models for Automatic Speech Recognition

Rohit Prabhavalkar and Tara Sainath
Tutorial T2, Interspeech 2018
http://interspeech2018.org/program-tutorials.html

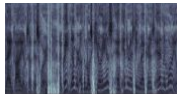
What is End-to-End ASR?

Conventional ASR

Pipeline



Input Speech



Feature Extraction

DNN/RNN Acoustic Model

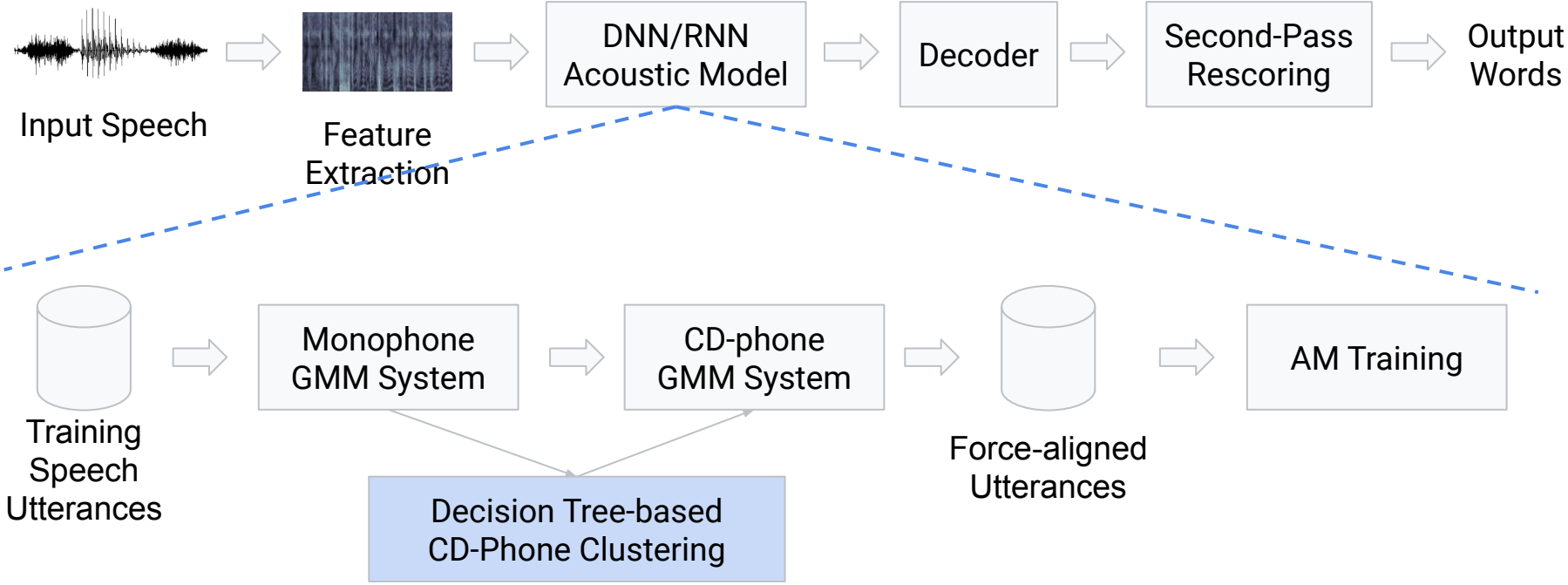
Decoder

Second-Pass Rescoring

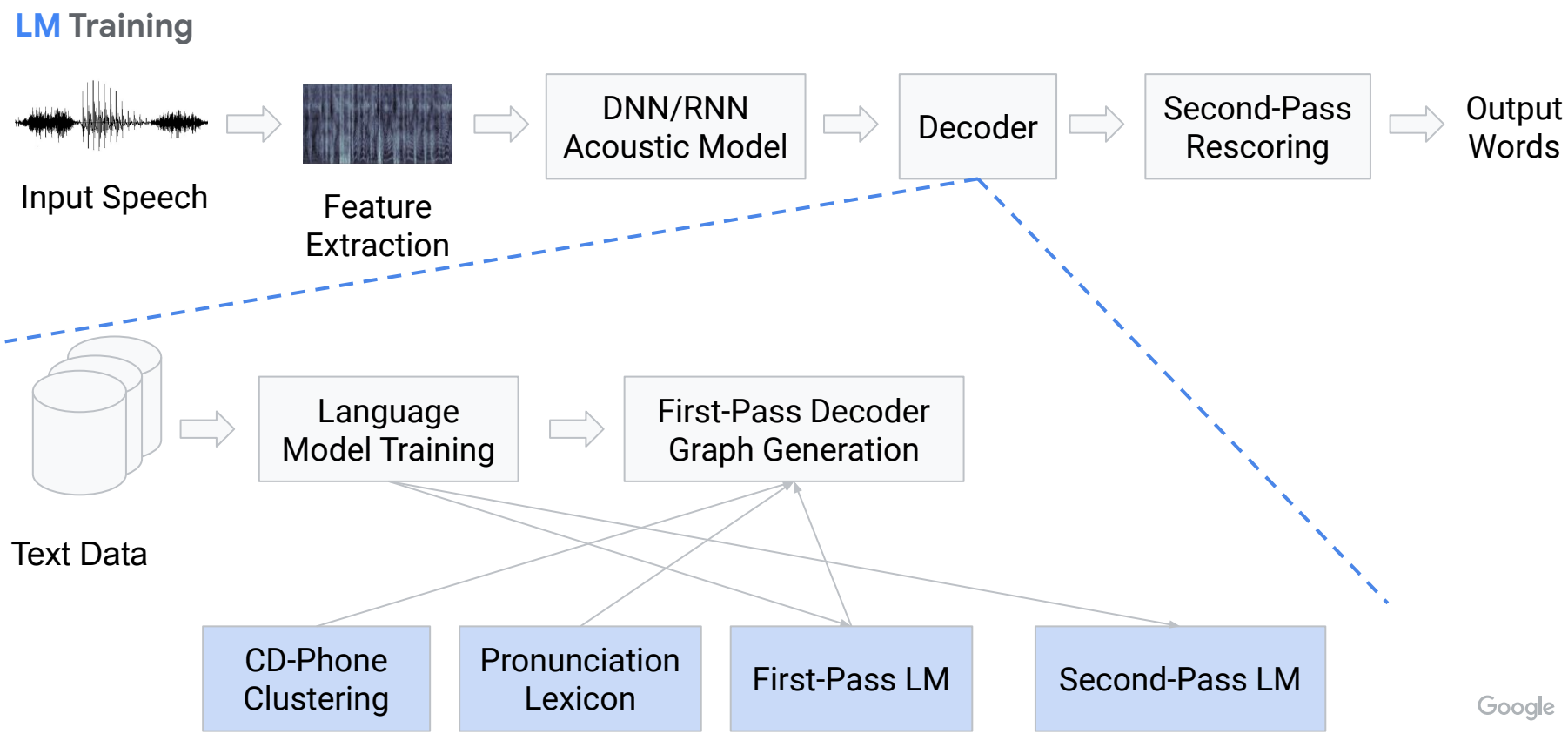
Output Words

Conventional ASR

AM Training



Conventional ASR



Conventional ASR

- Most ASR systems involve acoustic, pronunciation and language model components which are trained separately
  - Discriminative Sequence Training of AMs does couple these components
- Curating pronunciation lexicon, defining phoneme sets for the particular language requires expert knowledge, and is time-consuming

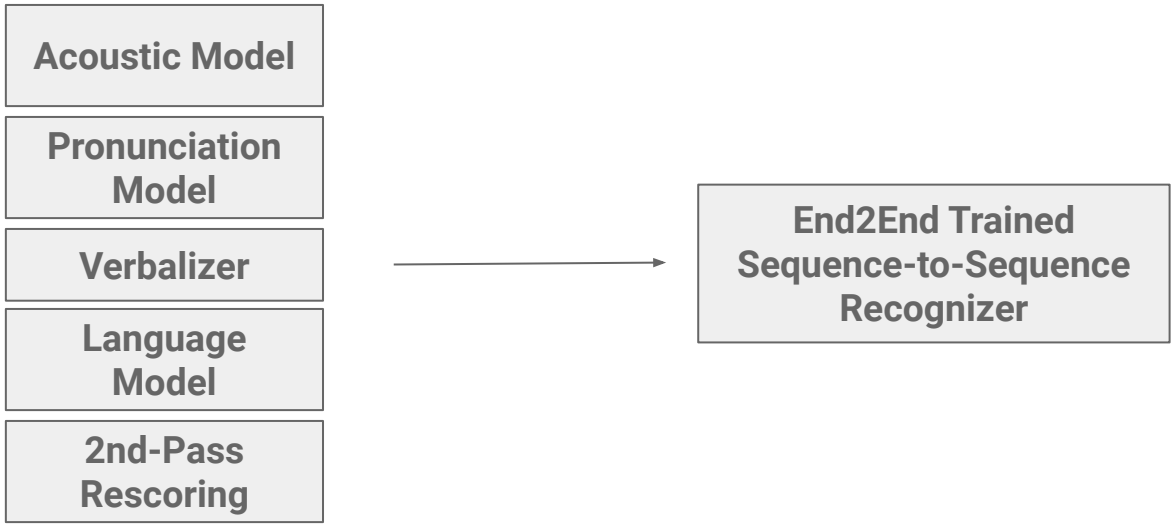
What is **End-to-End** ASR?

> A system which directly maps a sequence of input acoustic features into a sequence of graphemes or words.

> A system which is trained to optimize criteria that are related to the final evaluation metric that we are interested in (typically, word error rate).

Motivation

Typical Speech System



Key Takeaway

A single end-to-end trained sequence-to-sequence model, which directly outputs words or graphemes, could greatly simplify the speech recognition pipeline.

Agenda

Research developments on end-to-end models towards productionization.

Attention	Pushing the limit of attention-based end-to-end models.
Online Models	Streaming models for real world applications.
The Future	Future directions and challenges

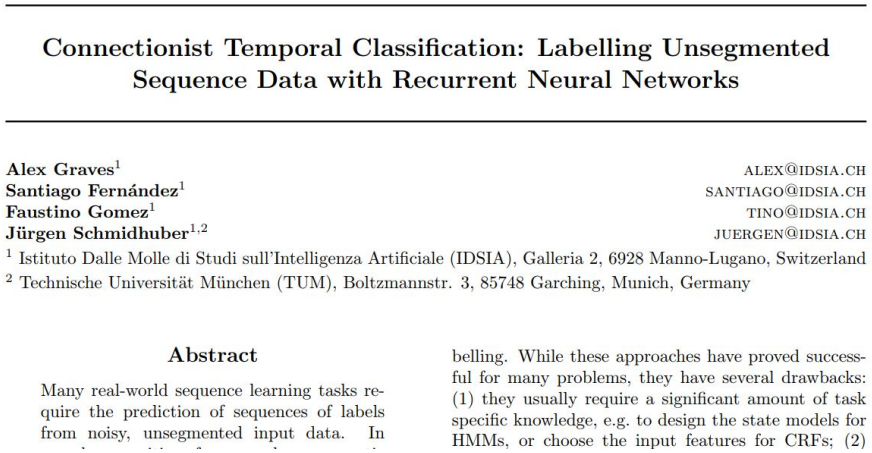
Historical Development of End-to-End ASR

CTC

Connectionist Temporal Classification

Connectionist Temporal Classification (CTC)

- CTC was proposed by [Graves et al., 2006] as a way to train an acoustic model without requiring frame-level alignments
- Early work, used CTC with phoneme output targets - not “end-to-end”
- CD-phoneme based CTC models achieve state-of-the-art performance for conventional ASR, but word-level lagged behind [Sak et al., 2015]



Connectionist Temporal Classification: Labelling Unsegmented Sequence Data with Recurrent Neural Networks

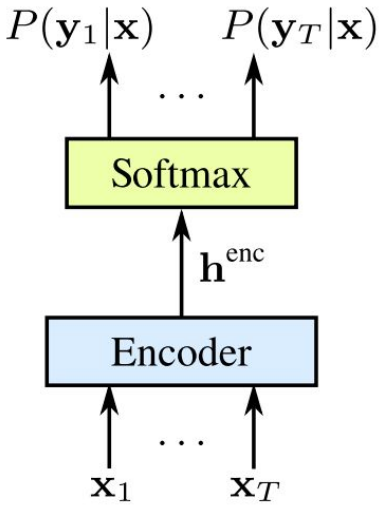
Alex Graves[1] ALEX@IDSIA.CH
Santiago Fernández[1] SANTIAGO@IDSIA.CH
Faustino Gomez[1] TINO@IDSIA.CH
Jürgen Schmidhuber[1,2] JUERGEN@IDSIA.CH
[1] Istituto Dalle Molle di Studi sull’Intelligenza Artificiale (IDSIA), Galleria 2, 6928 Manno-Lugano, Switzerland
[2] Technische Universität München (TUM), Boltzmannstr. 3, 85748 Garching, Munich, Germany

Abstract

Many real-world sequence learning tasks require the prediction of sequences of labels from noisy, unsegmented input data. In belling. While these approaches have proved successful for many problems, they have several drawbacks: (1) they usually require a significant amount of task specific knowledge, e.g. to design the state models for HMMs, or choose the input features for CRFs; (2)

[Graves et al., 2006] ICML

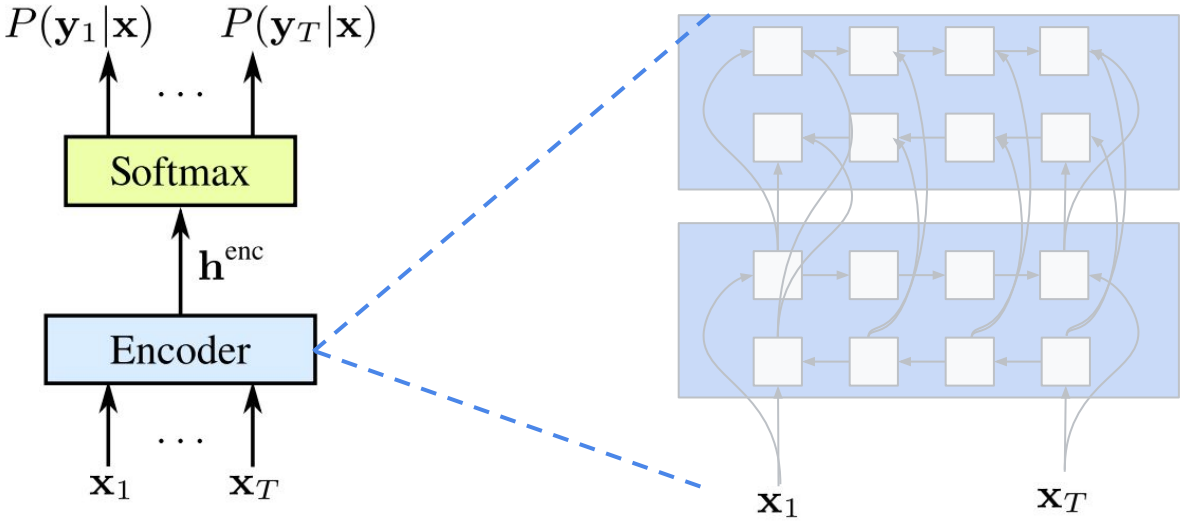
Connectionist Temporal Classification (CTC)



Key Takeaway

CTC allows for training an acoustic model without the need for frame-level alignments between the acoustics and the transcripts.

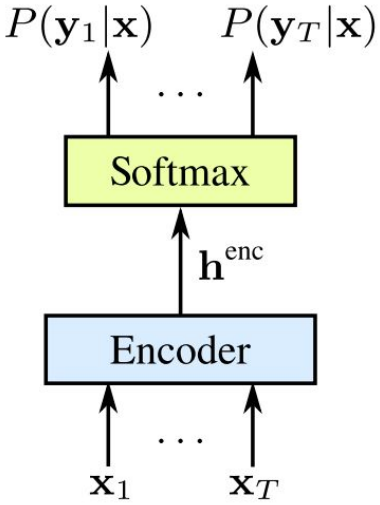
Connectionist Temporal Classification (CTC)



Key Takeaway

Encoder: Multiple layers of Uni- or Bi-directional RNNs (often LSTMs).

Connectionist Temporal Classification (CTC)



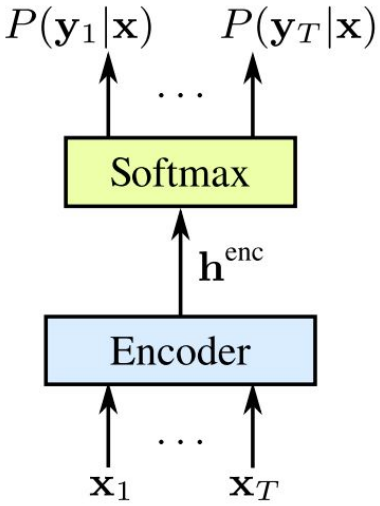
B	B	**c**	B	B	**a**	**a**	B	B	**t**
B	**c**	**c**	B	**a**	B	B	B	B	**t**
				...					
B	**c**	B	B	**a**	B	B	**t**	**t**	B

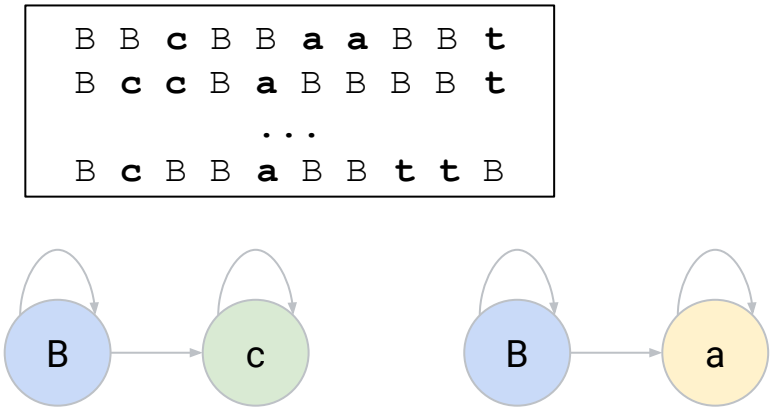
$$P(\mathbf{y}|\mathbf{x}) = \sum_{\hat{\mathbf{y}} \in \mathcal{B}(\mathbf{y},\mathbf{x})} \prod_{t=1}^{T} P(\hat{y}_t|\mathbf{x})$$

Key Takeaway

CTC introduces a special symbol - blank (denoted by B) - and maximizes the total probability of the label sequence by marginalizing over all possible alignments

Connectionist Temporal Classification (CTC)

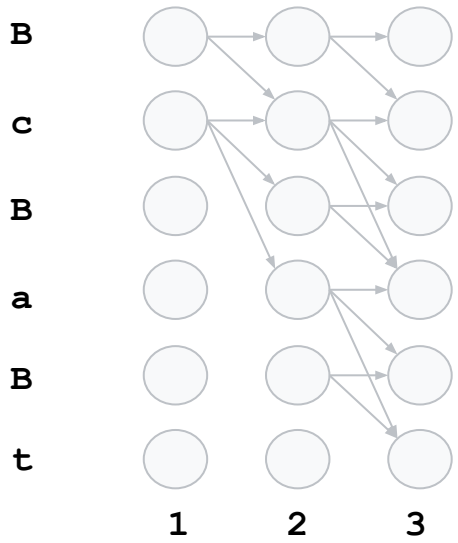




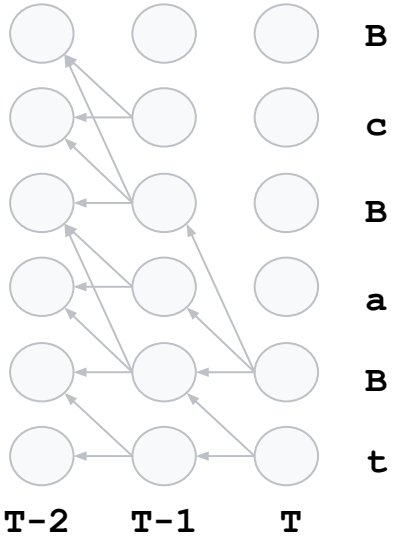
Key Takeaway

In a conventional hybrid system, this would correspond to defining the HMMs corresponding to each unit to consist of a shared initial state (blank), followed by a separate state(s) for the actual unit.

Connectionist Temporal Classification (CTC)



Forward-Backward Algorithm Computation



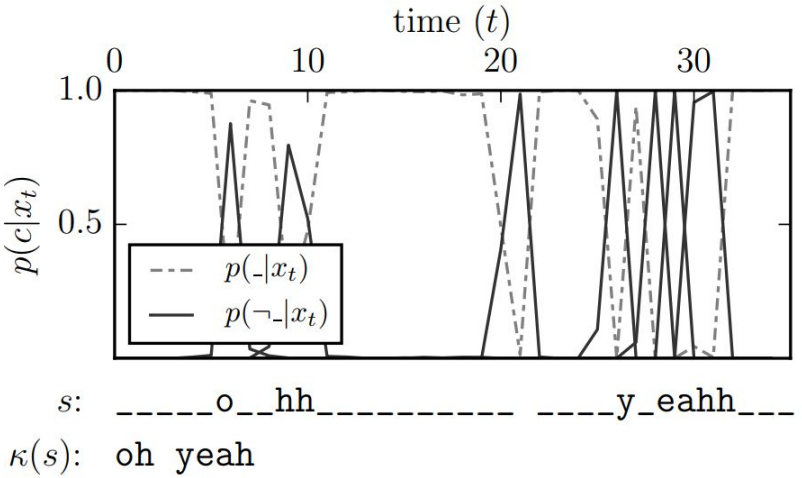
Frames, t

Key Takeaway

Computing the gradients of the loss requires the computation of the alpha-beta variables using the forward-backward algorithm [Rabiner, 1989]

CTC-Based End-to-End ASR

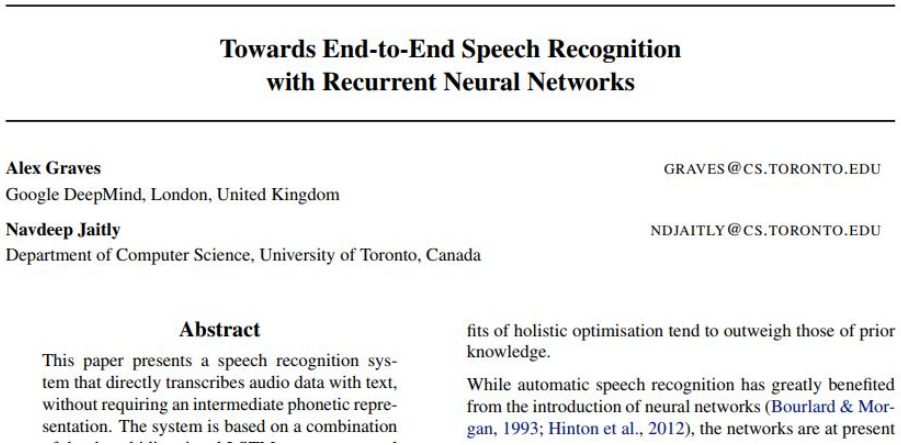
CTC produces “spiky” and sparse activations - can sometimes directly read off the final transcription from the activations even without an LM



Reproduced from [Maas et al., 2015] NAACL

CTC-Based End-to-End ASR

- Graves and Jaitly proposed a system with character-based CTC which directly output word sequences given input speech
- Using an external LM was important for getting good performance. Results reported by rescoring a baseline system.
- Also proposed minimizing expected transcription error [WSJ: 8.7% → 8.2%]



Towards End-to-End Speech Recognition with Recurrent Neural Networks

Alex Graves GRAVES@CS.TORONTO.EDU
Google DeepMind, London, United Kingdom

Navdeep Jaitly NDJAITLY@CS.TORONTO.EDU
Department of Computer Science, University of Toronto, Canada

Abstract

This paper presents a speech recognition system that directly transcribes audio data with text, without requiring an intermediate phonetic representation. The system is based on a combination

fits of holistic optimisation tend to outweigh those of prior knowledge.

While automatic speech recognition has greatly benefited from the introduction of neural networks (Bourlard & Morgan, 1993; Hinton et al., 2012), the networks are at present

[Graves and Jaitly, 2014] ICML

CTC-Based ASR

Refinements since [Graves & Jaitly, 2014]

- LM incorporated into first-pass decoding; easy integration with WFSTs
  - [Hannun et al., 2014] [Maas et al., 2015]: Direct first-pass decoding with an LM as opposed to rescoring as in [Graves & Jaitly, 2014]
  - [Miao et al., 2015]: EESEN framework for decoding with WFSTs, open source toolkit
- Large-scale GPU training; data augmentation; multiple languages
  - [Hannun et al., 2014; DeepSpeech] [Amodei et al., 2015; DeepSpeech2]: Large scale GPU training; Data Augmentation; Mandarin and English
- Using longer span units: words instead of characters
  - [Soltau et al., 2017]: Word-level CTC targets, trained on 125,000 hours of speech. Performance close to or better than a conventional system, even without using an LM!
  - [Audhkhasi et al., 2017]: Direct Acoustics-to-Word Models on Switchboard
- And many others ...

Shortcomings of CTC

- For efficiency, CTC makes an important independence assumption - network outputs at different frames are conditionally independent
- Obtaining good performance from CTC models requires the use of an external language model - direct greedy decoding does not perform very well

Attention-based Encoder-Decoder Models

Attention-based Encoder-Decoder Models

- Attention-based Encoder-Decoder Models emerged first in the context of neural machine translation.
- Were first applied to ASR by [Chan et al., 2015] [Chorowski et al., 2015]

Listen, Attend and Spell

William Chan
Carnegie Mellon University
williamchan@cmu.edu

Navdeep Jaitly, Quoc V. Le, Oriol Vinyals
Google Brain
{ndjaitly,qvl,vinyals}@google.com

[Chan et al., 2015]

Attention-Based Models for Speech Recognition

Jan Chorowski
University of Wrocław, Poland
jan.chorowski@ii.uni.wroc.pl

Dzmitry Bahdanau
Jacobs University Bremen, Germany

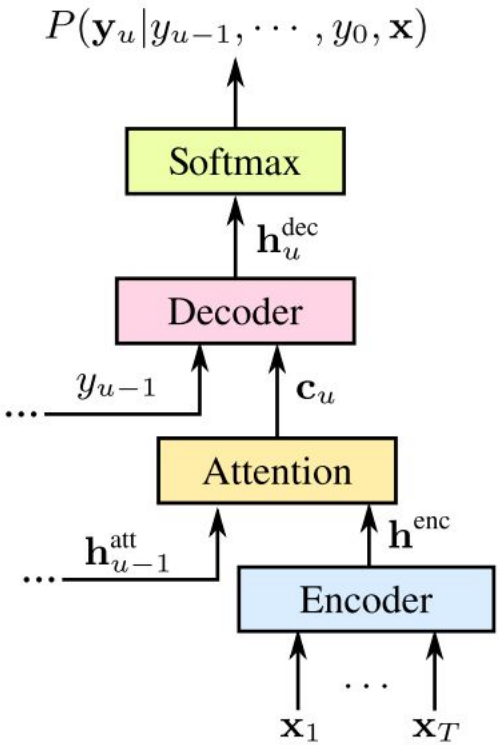
Dmitriy Serdyuk
Université de Montréal

Kyunghyun Cho
Université de Montréal

Yoshua Bengio
Université de Montréal
CIFAR Senior Fellow

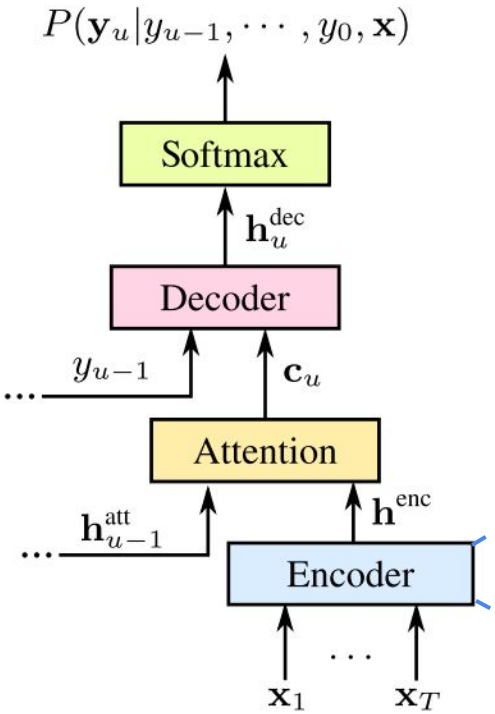
[Chorowski et al., 2015]

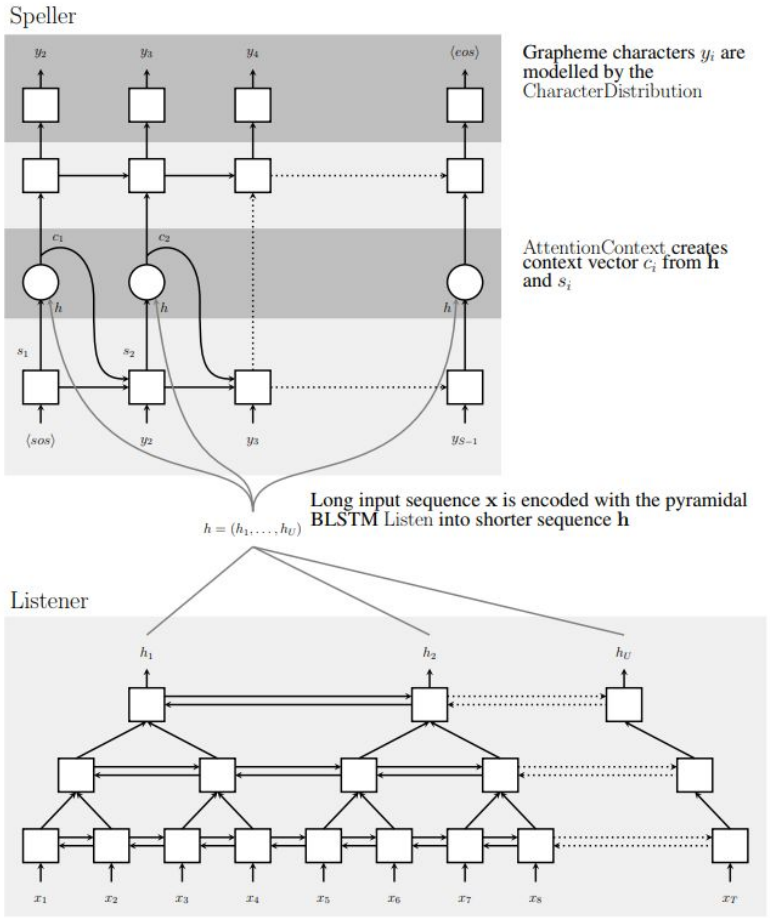
Attention-based Encoder-Decoder Models



- Encoder (analogous to AM):
  - Transforms input speech into higher-level representation
- Attention (alignment model):
  - Identifies encoded frames that are relevant to producing current output
- Decoder (analogous to PM, LM):
  - Operates autoregressively by predicting each output token as a function of the previous predictions

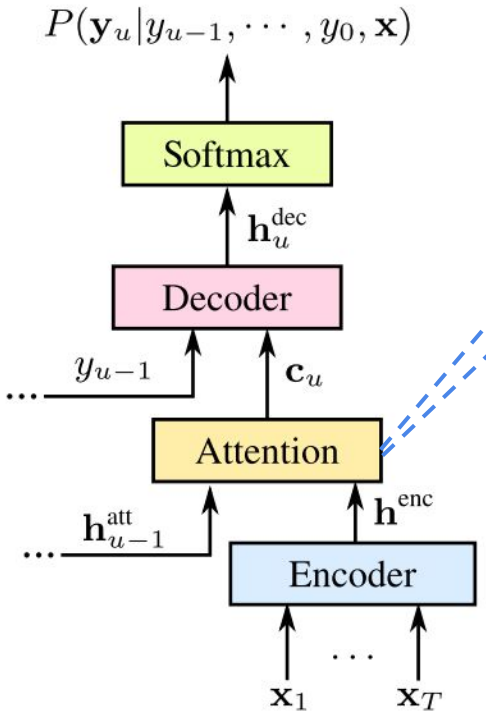
Attention-based Models

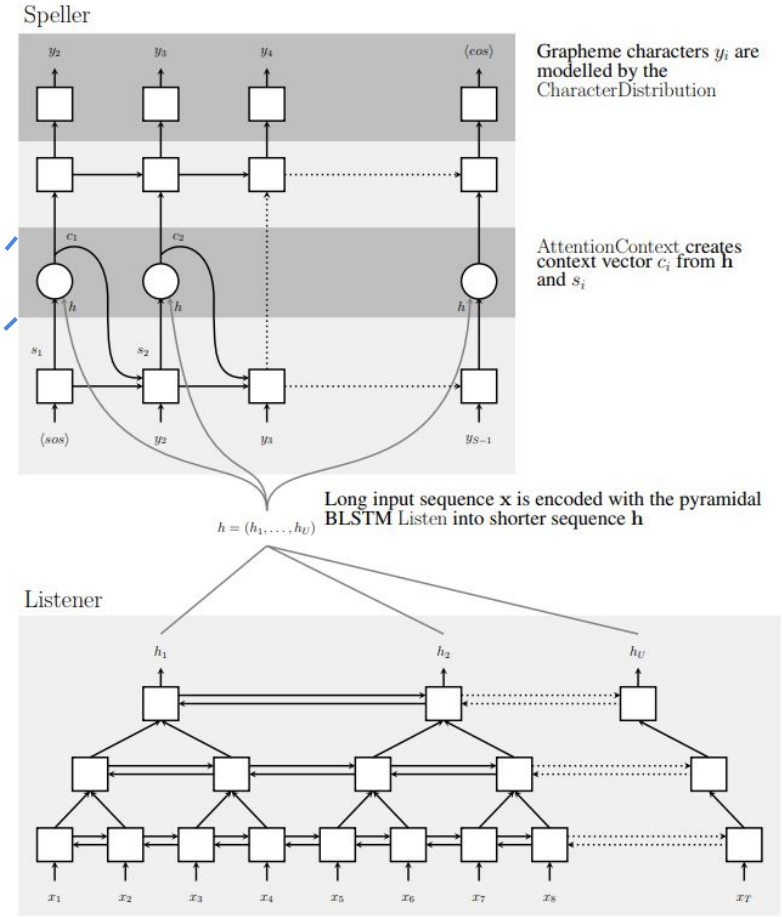




Reproduced from [Chan et al., 2015]

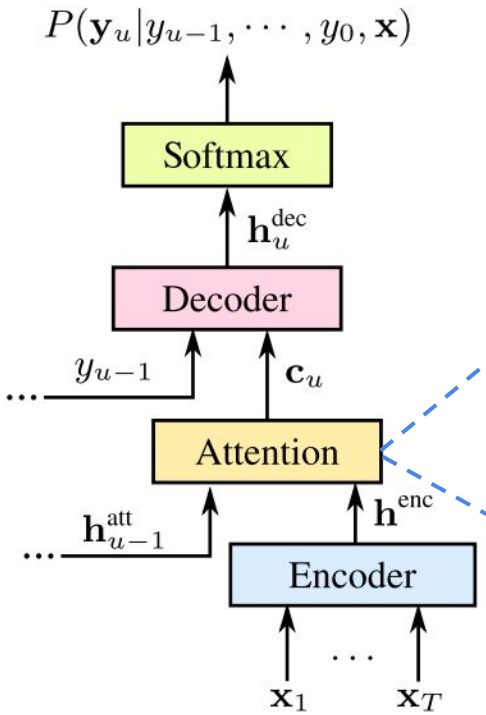
Attention-based Models





Reproduced from [Chan et al., 2015]

Attention-based Models



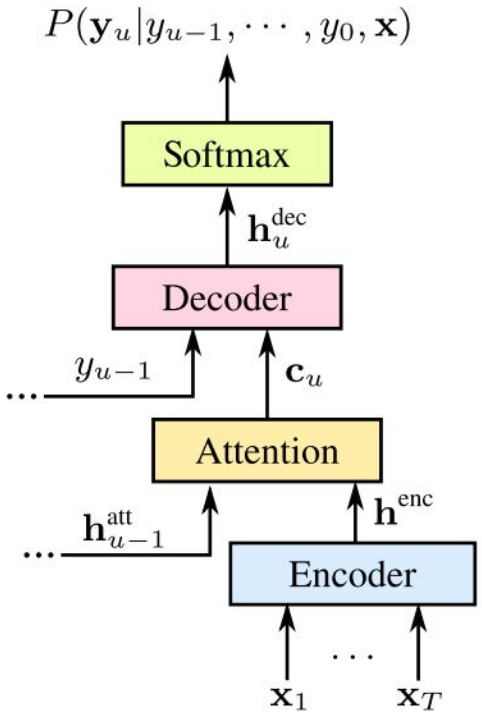
Attention module computes a similarity score between the decoder and each frame of the encoder

$$e_{u,t} = \text{score}(\mathbf{h}_{u-1}^{\text{att}}, \mathbf{h}_t^{\text{enc}})$$

$$\alpha_{u,t} = \frac{\exp(e_{u,t})}{\sum_{t'=1}^{T} \exp(e_{u,t'})}$$

$$\mathbf{c}_u = \sum_{t=1}^{T} \alpha_{u,t} h_t^{\text{enc}}$$

Attention-based Models



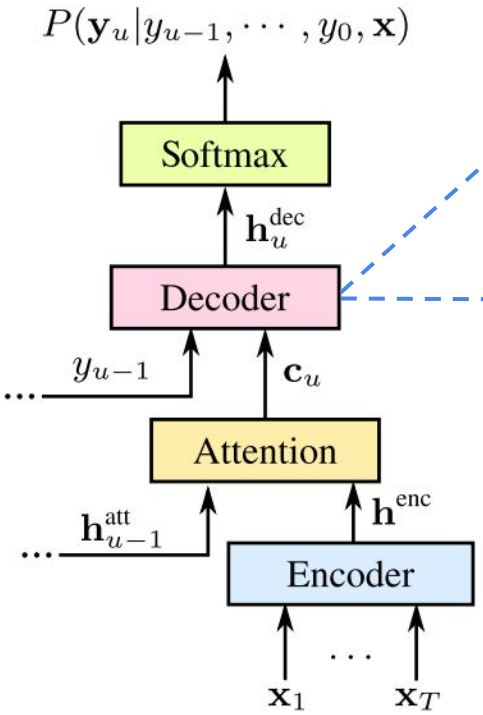
Dot-Product Attention [Chan et al., 2015]

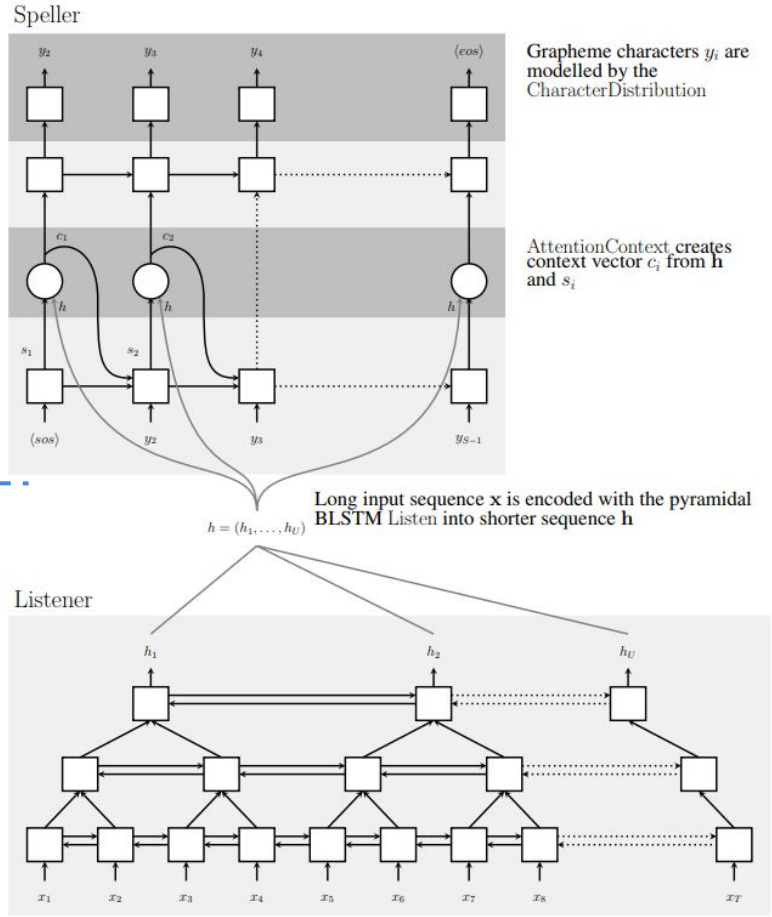
$$e_{u,t} = \left\langle \phi(W\mathbf{h}_{u-1}^{\text{att}}),\ \psi(V\mathbf{h}_t^{\text{enc}}) \right\rangle$$

Additive Attention [Chorowski et al., 2015]

$$e_{u,t} = w^T \tanh(W\mathbf{h}_{u-1}^{\text{att}} + V\mathbf{h}_t^{\text{enc}} + b)$$

Attention-based Models



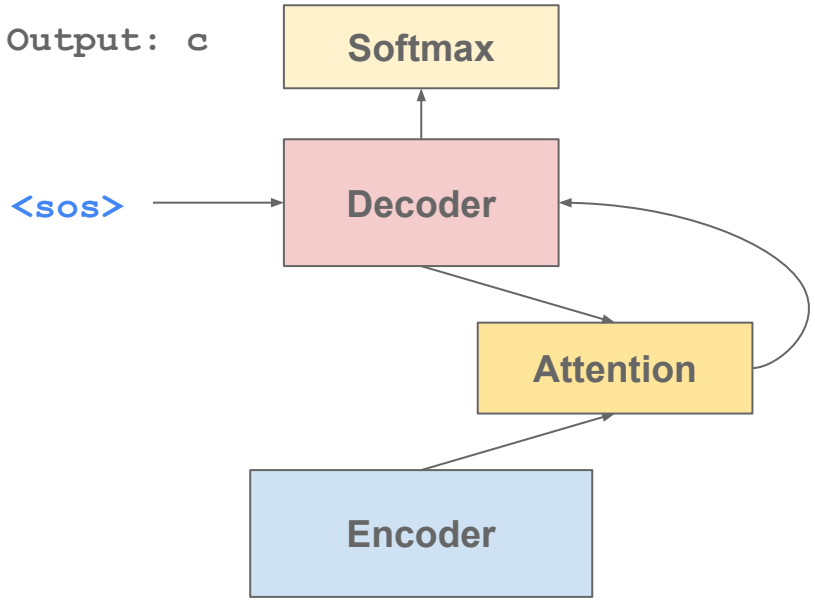


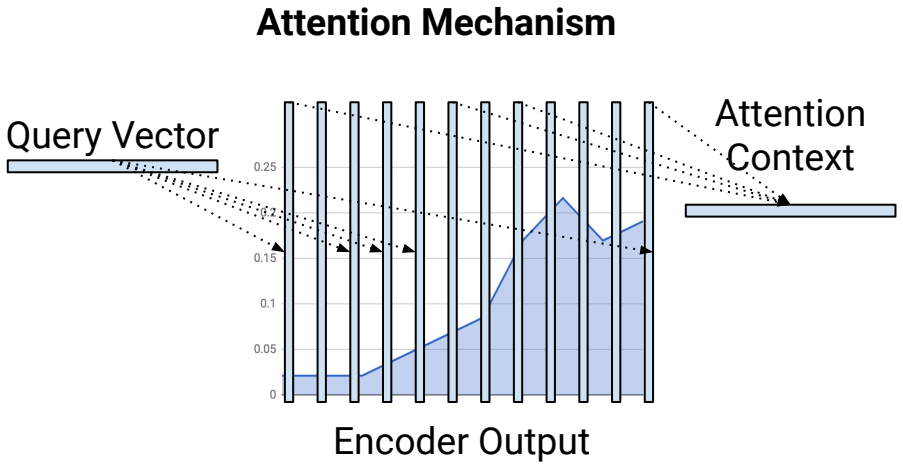
Reproduced from [Chan et al., 2015]

Attention-based Models

```
P(a|<sos>,x) = 0.01
P(b|<sos>,x) = 0.01
P(c|<sos>,x) = 0.92
       ...
```

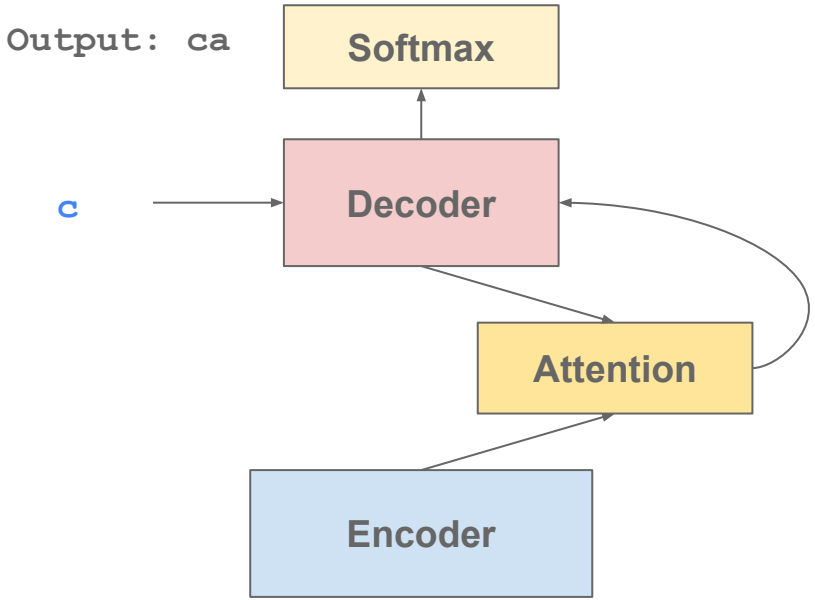
Attention mechanism summarizes encoder features relevant to predict next label





Attention-based Models

```
P(a|c,<sos>,x) = 0.95
P(b|c,<sos>,x) = 0.01
P(c|c,<sos>,x) = 0.01
          ...
```

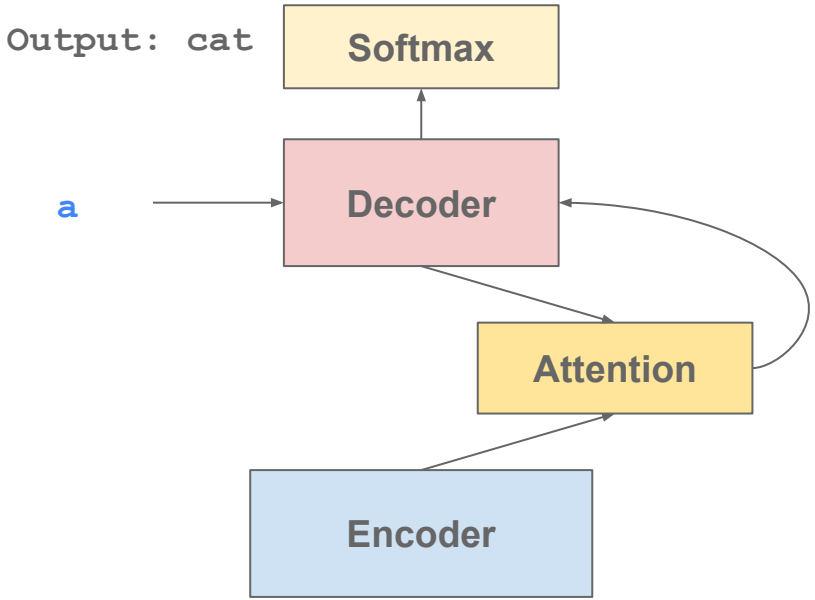


Labels from previous step are fed into decoder at the next step to predict

$$P(\mathbf{y}_u | y_{u-1}, \cdots, y_0, \mathbf{x})$$

Attention-based Models

```
P(a|a,c,<sos>,x) = 0.01
P(b|a,c,<sos>,x) = 0.08
              ...
P(t|a,c,<sos>,x) = 0.89
              ...
```

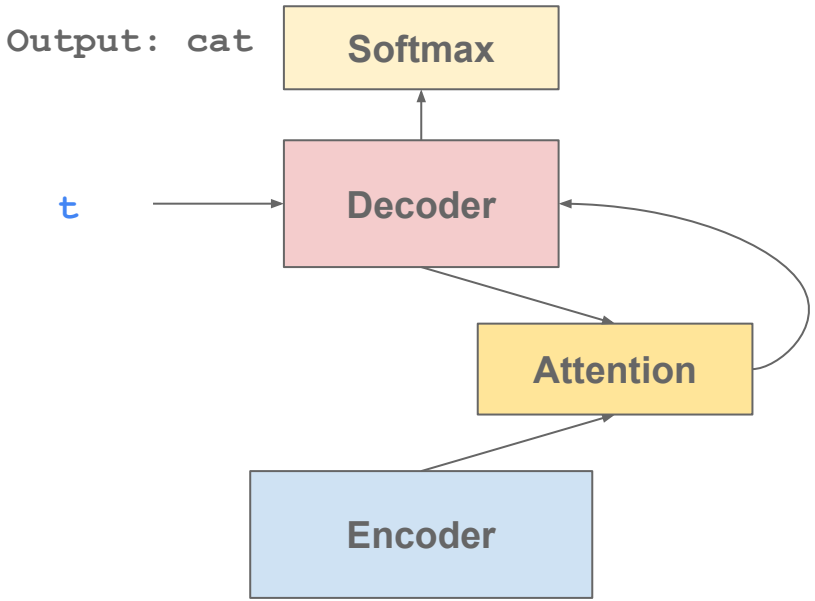


Labels from previous step are fed into decoder at the next step to predict

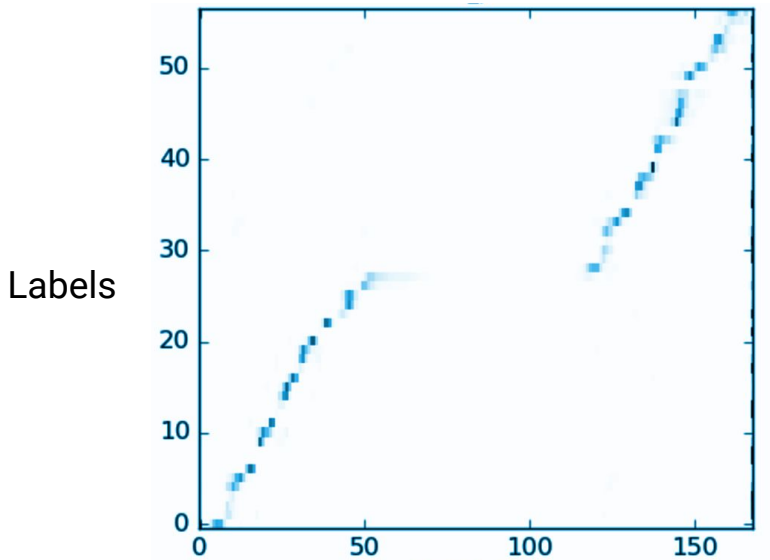
$$P(\mathbf{y}_u | y_{u-1}, \cdots, y_0, \mathbf{x})$$

Attention-based Models

```
P(a|t,a,c,<sos>,x) = 0.01
P(b|t,a,c,<sos>,x) = 0.01
                ...
P(<eos>|t,a,c,<sos>,x) = 0.96
                ...
```



Process terminates when the model predicts <eos> which denotes end of sentence.

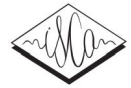


Model Comparisons on a 12,500 hour Google Task

Comparing Various End-to-End Approaches

- Compare various sequence-to-sequence models head-to-head, trained on same data, to understand how these approaches compare to each other
- Evaluated on a large-scale 12,500 hour Google Voice Search Task

INTERSPEECH 2017
August 20–24, 2017, Stockholm, Sweden



A Comparison of Sequence-to-Sequence Models for Speech Recognition

Rohit Prabhavalkar[1], *Kanishka Rao*[1], *Tara N. Sainath*[1], *Bo Li*[1], *Leif Johnson*[1], *Navdeep Jaitly*[2†]

[1]Google Inc., U.S.A
[2]NVIDIA, U.S.A.

{prabhavalkar,kanishkarao,tsainath,boboli,leif}@google.com, njaitly@nvidia.com

Abstract

In this work, we conduct a detailed evaluation of various all-neural, end-to-end trained, sequence-to-sequence models applied to the task of speech recognition. Notably, each of these

was shown to outperform a state-of-the-art CD-phoneme baseline on a YouTube video captioning task. The basic CTC model was extended by Graves [3] to include a separate recurrent language model component, in a model referred to as the recurrent neural network (RNN) transducer. Although this model has

[Prabhavalkar et al., 2017]

Experimental Setup

Model Configurations

- Baseline
  - State-of-the-art CD-Phoneme model: 5x700 BLSTM; ~8000 CD-Phonemes
  - CTC-training followed by sMBR discriminative sequence training
  - Decoded with large 5-gram LM in first pass
  - Second pass rescoring with much larger 5-gram LM
  - Lexicon of millions of words of expert curated pronunciations
- Sequence-to-Sequence Models
  - Trained to output graphemes: [a-z], [0-9], <space>, and punctuation
  - Models are evaluated using beam search (Keep Top 15 Hyps at Each Step)
  - ***Models are not decoded or rescored with an external language model, or a pronunciation model***

Experimental Setup

Data

- Training Set
  - ~15M Utterances (~12,500 hrs) of anonymized utterances from Google Voice Search Traffic
  - Multi-style Training: Artificially distorted using room simulator by adding noise samples extracted from YouTube videos and environmental recordings of daily events
- Evaluation Sets
  - Dictation: ~13K utterances (~124K words) open-ended dictation
  - VoiceSearch: ~12.9K utterances (~63K words) of voice-search queries

Results

Model	Clean	
	Dictation	VoiceSearch
Baseline Uni. Context Dependent Phones (CDP)	6.4	9.9
Baseline BiDi. CDP	**5.4**	**8.6**
CTC-grapheme	39.4	53.4

Key Takeaway

Decoding CTC-grapheme models without an LM performs poorly.

Results

Model	Clean	
	Dictation	VoiceSearch
Baseline Uni. CDP	6.4	9.9
Baseline BiDi. CDP	**5.4**	**8.6**
CTC-grapheme	39.4	53.4
Attention-based Model	6.6	11.7

Key Takeaway

Attention-based model performs the best, but still lags behind a conventional model.

Further Improvements

Improvements for productions

[Chiu et al., 2018]

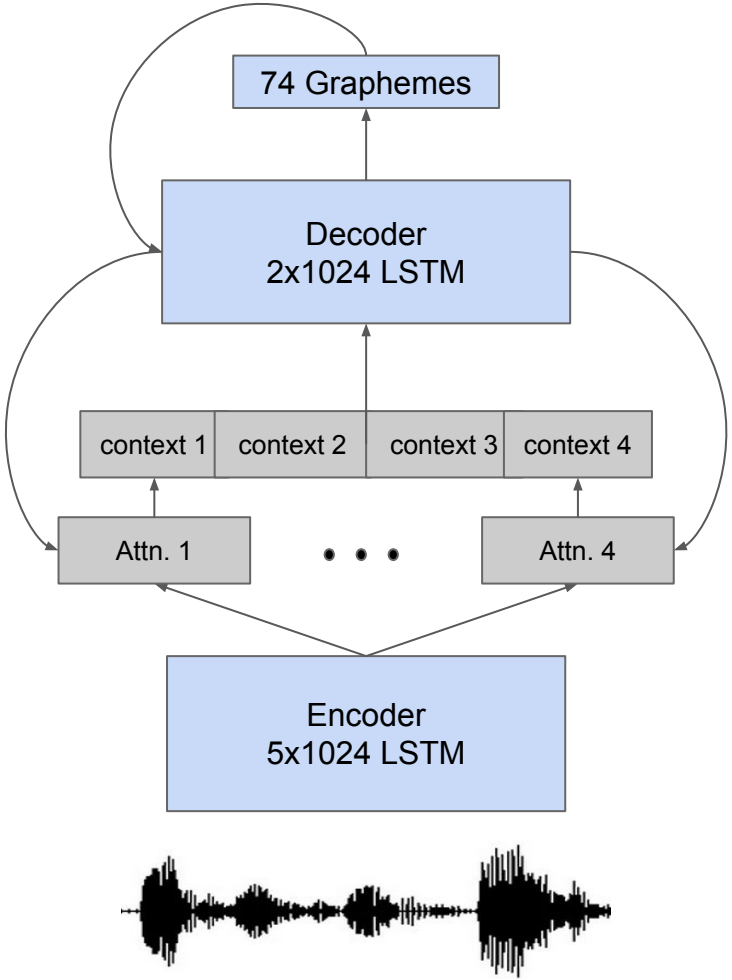
To match an end-to-end model to a strong conventional system:

- Structural improvements
  - Wordpiece models
  - Multi-headed attention
- Optimization improvements
  - Minimum word error rate (MWER) training
  - Scheduled sampling
  - Asynchronous and synchronous training
  - Label smoothing

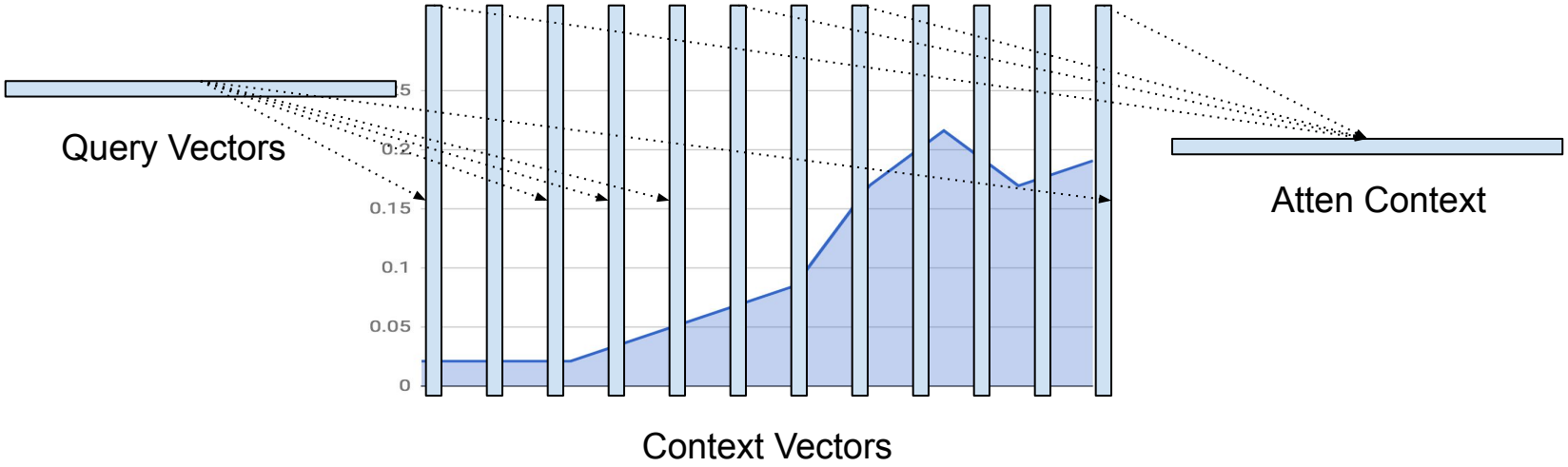
Structure improvements

(1) Improved Attention

- Encoder network consists of 5 unidirectional-directional LSTM layers
- Decoder network consists of 2 LSTM layers
- Decode into graphemes directly (74 targets)
- This work: explore *Multi-headed attention* with 4 parallel attention heads [A. Vaswani, 2017]

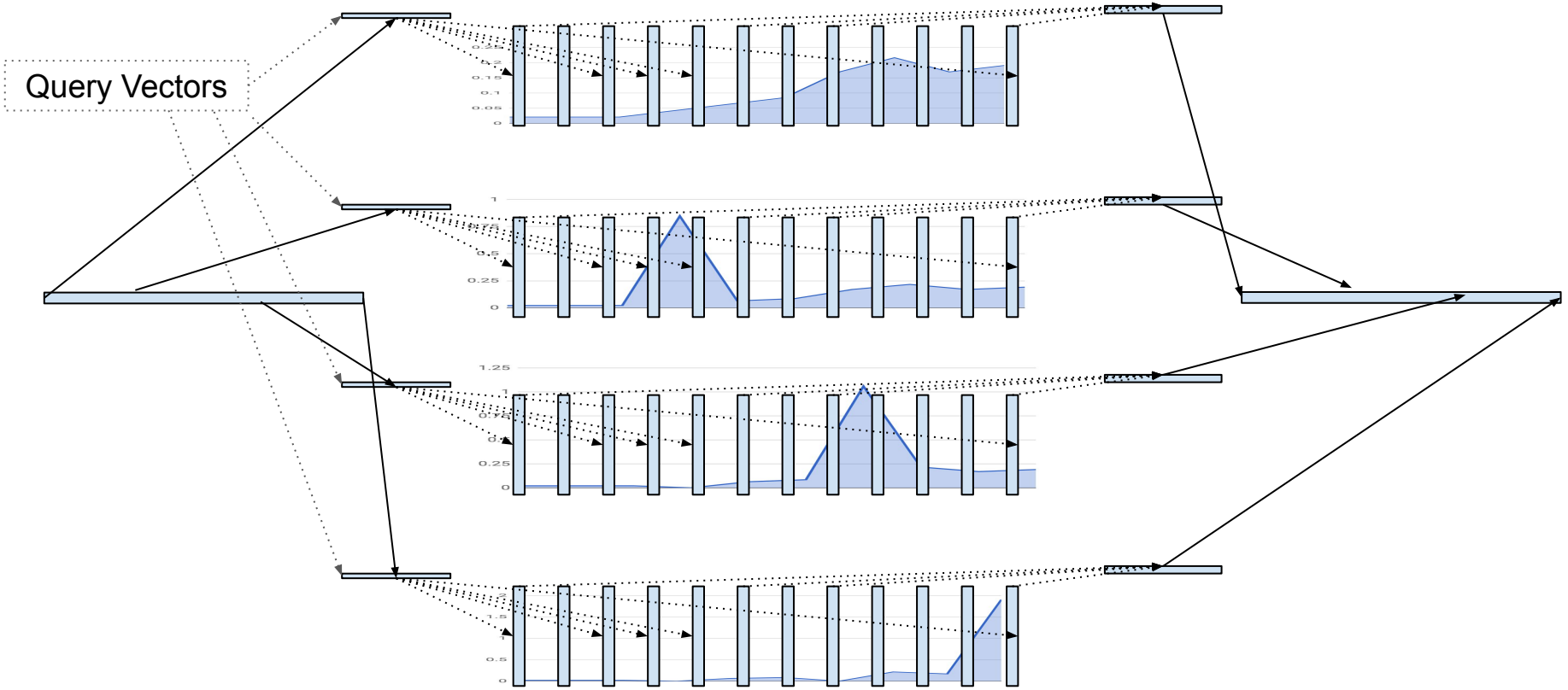


Single headed attention

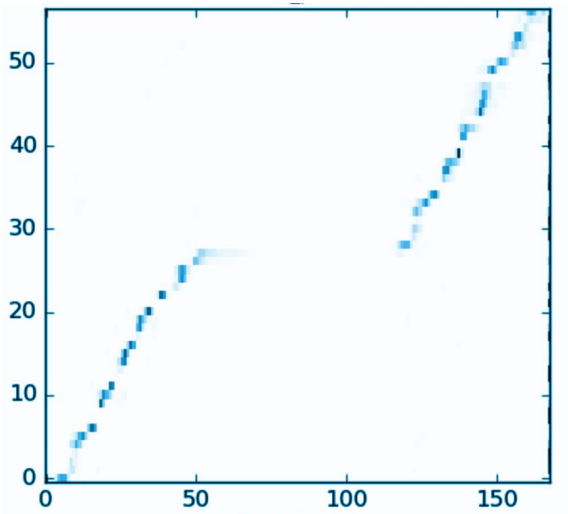


Attention Context is the weighted average of context vectors

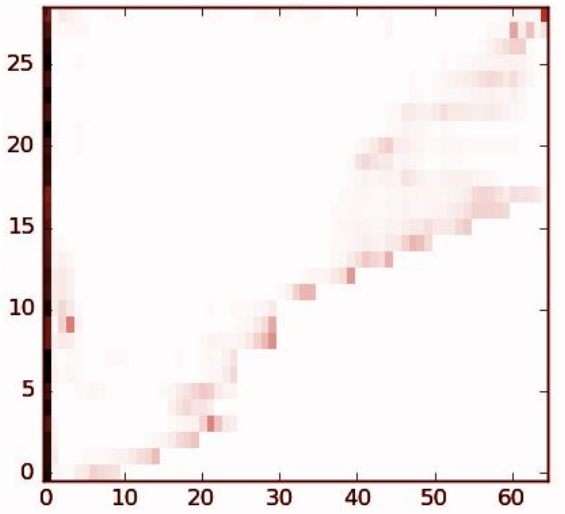
Multi-headed attention



Single vs. Multi-headed Attention Visualized



Single Headed Attention



Multi-Headed Attention

Multi-headed attention examines different parts of the utterance for each predicted label.
Model looks predominantly towards previous frames

(2) Word Pieces

- We want to use subword units longer than graphemes:
  - Longer units have a lower LM perplexity
  - Longer units gives improved decoder efficiency
- Word pieces is a good longer-unit choice [Schuster, 2012]
  - Has shown good results for RNN-T [Rao, ASRU 2017]
- Word piece model (WPM) details
  - Trained to maximize LM likelihood on training data
  - Position dependent, determined determinstically
  - Units back off to characters → No OOVs

Good Afternoon → _go o d _aft er noon

Optimization improvements

(3) Minimum Word Error Rate (MWER)

[Prabhavalkar et al., 2018]

- Attention-based Sequence-to-Sequence models are typically trained by optimizing cross entropy loss (i.e., maximizing log-likelihood of the training data)

$$\mathcal{L}_{\text{CE}} = \sum_{(\mathbf{x},\mathbf{y}^*)} \sum_{u=1}^{L+1} -\log P(y_u^* | y_{u-1}^*, \cdots, y_0^* = \langle \text{sos} \rangle, \mathbf{x})$$

- Training criterion does not match metric of interest: Word Error Rate
- Goal: Optimize a loss that minimizes or is correlated with minimizing word error rate

Minimum Word Error Rate (MWER)

$$\mathcal{L}_{\text{werr}}(\mathbf{x}, \mathbf{y}^*) = \mathbb{E}[\mathcal{W}(\mathbf{y}, \mathbf{y}^*)] = \sum_{\mathbf{y}} P(\mathbf{y}|\mathbf{x})\mathcal{W}(\mathbf{y}, \mathbf{y}^*)$$

Number of Word Errors

Key Takeaway

Minimizing expected WER directly is intractable since it involves a summation over all possible label sequences. Approximate expectation using samples.

Minimum Word Error Rate (MWER)

- Approximate expectation using samples [Shannon, 2017].

$$\mathcal{L}_{\text{werr}}(\mathbf{x}, \mathbf{y}^*) \approx \mathcal{L}_{\text{werr}}^{\text{Sample}}(\mathbf{x}, \mathbf{y}^*) = \frac{1}{N} \sum_{\mathbf{y}_i \sim P(\mathbf{y}|\mathbf{x})} \mathcal{W}(\mathbf{y}_i, \mathbf{y}^*)$$

- Approximation using N-Best List [Stolcke et al., 1997][Povey, 2003]

$$\mathcal{L}_{\text{werr}}^{\text{N-best}}(\mathbf{x}, \mathbf{y}^*) = \sum_{\mathbf{y}_i \in \text{Beam}(\mathbf{x}, N)} \widehat{P}(\mathbf{y}_i|\mathbf{x}) \left[\mathcal{W}(\mathbf{y}_i, \mathbf{y}^*) - \widehat{W} \right]$$

Approximation using N-Best List [Stolcke+,97][Povey,03]

$$\mathcal{L}_{\text{werr}}(\mathbf{x}, \mathbf{y}^*) = \mathbb{E}[\mathcal{W}(\mathbf{y}, \mathbf{y}^*)] = \sum_{\mathbf{y}} P(\mathbf{y}|\mathbf{x})\mathcal{W}(\mathbf{y}, \mathbf{y}^*)$$

$$\mathcal{L}^{\text{N-best}}_{\text{werr}}(\mathbf{x}, \mathbf{y}^*) = \sum_{\mathbf{y}_i \in \text{Beam}(\mathbf{x}, N)} \widehat{P}(\mathbf{y}_i|\mathbf{x}) \left[\mathcal{W}(\mathbf{y}_i, \mathbf{y}^*) - \widehat{W}\right]$$

$$\widehat{P}(\mathbf{y}_i|\mathbf{x}) = \frac{P(\mathbf{y}_i|\mathbf{x})}{\sum_{\mathbf{y}_i \in \text{Beam}(\mathbf{x}, N)} P(\mathbf{y}_i|\mathbf{x})}$$

Assume that probability distribution is concentrated on top-N hypotheses

(4) Some more optimization improvements

- Scheduled Sampling [S. Bengio, 2015]
  - Feedback in the prediction from the model rather than the true previous prediction
  - Helps prevent overfitting
- Label smoothing [C. Szegedy, 2016]
  - Take the logit class with maximum probability and smooth it over the remaining labels
  - Helps prevent overfitting
- Sync training [P. Goyal, 2017]
  - Gradient updates between workers are synchronized
  - Leads to faster convergence and better model quality

Experimental Details

Training data:

- 15M English utterances
- 12,500 hours noisy data
- artificially corrupted with music, ambient noise, recordings of "daily life" environments
- SNRs: 0 ~ 30dB, avg. = 11dB

Testing data:

- 14.8K English utterances
- 15 hours data
- simulated: matching training data

Model architecture:

- 80-dim lmel features, stacked and downsampled to 30ms
- Encoder: 5x1400 unidirectional LSTM
- Decoder: 2x1024
- 16K Word Pieces
- 4 attention heads

Results With Different Improvements

Exp ID	Model	WER - VS	WERR
E1	Grapheme	9.2	-
E2	WPM	9.0	2.2%
E3	+MHA	8.0	11.1%
E4	+Optimization*	6.7	16%
E5	MWER	5.8	**13.4%**

C.C. Chiu, et al, "State-of-the-art Speech Recognition With Sequence-to-Sequence Models," ICASSP, 2018.

* Includes sync training, label smoothing and scheduled sampling

Comparison to Conventional Model

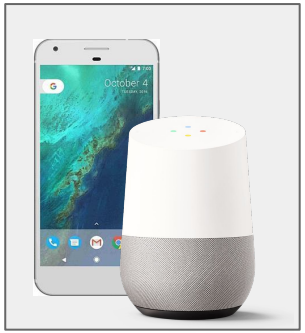
System	1st Pass Model Size	VS	IME
Conventional Baseline Model	0.1 GB (AM) + 2.2 GB (CL.fst) + 4.9 GB (G.fst) = 7.2GB, total	6.7	5.0
LAS	0.4 GB, total	**5.6 (-16%)**	**4.1 (-18%)**

- **16-18% rel improvement** over production
- **18X smaller** than conventional model in 1st pass
- Main drawback: model is not streaming

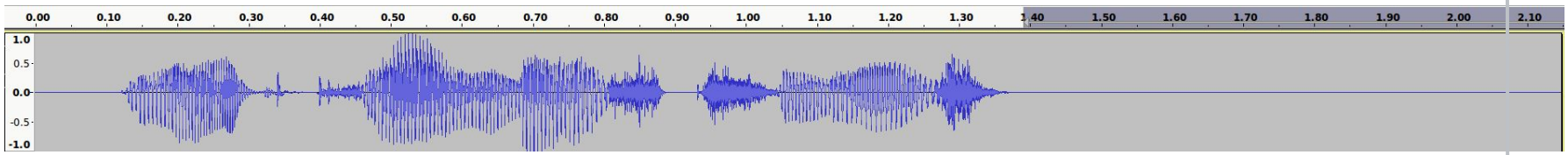
Online Models

RNN-T

Streaming Speech Recognition



Finalize recognition &
Taking action / fetching the search results



Recognize the audio

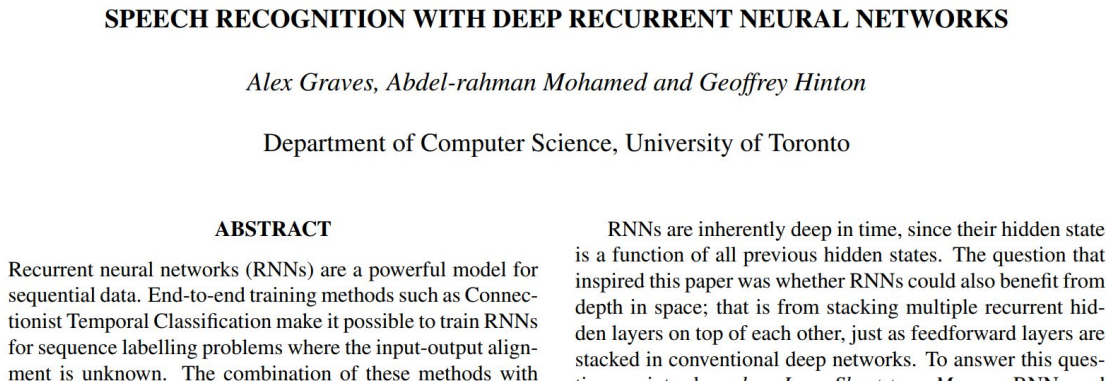
Online Models

- LAS is not streaming
- There are a variety of different online models
  - **RNN-T [Graves, 2012], [Rao et al., 2017], [He et al., 2018]**
  - Neural Transducer [Jaitly et al., 2015], [Sainath et al., 2018]
  - MoChA [Chiu and Raffel, 2018]

Recurrent Neural Network Transducer

Recurrent Neural Network Transducer (RNN-T)

- Proposed by Graves et al., RNN-T augments a CTC-based model with a recurrent LM component
- Both components are trained jointly on the available acoustic data
- As with CTC, the method does not require aligned training data.



SPEECH RECOGNITION WITH DEEP RECURRENT NEURAL NETWORKS

Alex Graves, Abdel-rahman Mohamed and Geoffrey Hinton

Department of Computer Science, University of Toronto

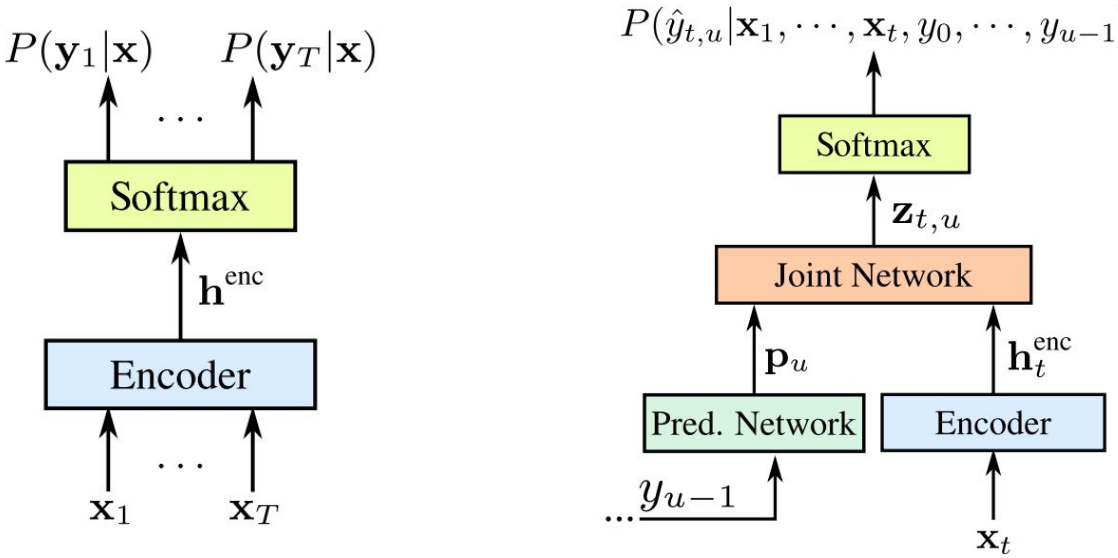
ABSTRACT

Recurrent neural networks (RNNs) are a powerful model for sequential data. End-to-end training methods such as Connectionist Temporal Classification make it possible to train RNNs for sequence labelling problems where the input-output alignment is unknown. The combination of these methods with

RNNs are inherently deep in time, since their hidden state is a function of all previous hidden states. The question that inspired this paper was whether RNNs could also benefit from depth in space; that is from stacking multiple recurrent hidden layers on top of each other, just as feedforward layers are stacked in conventional deep networks. To answer this ques-

[Graves et al., 2013] ICASSP;
[Graves, 2012] ICML Representation Learning Workshop

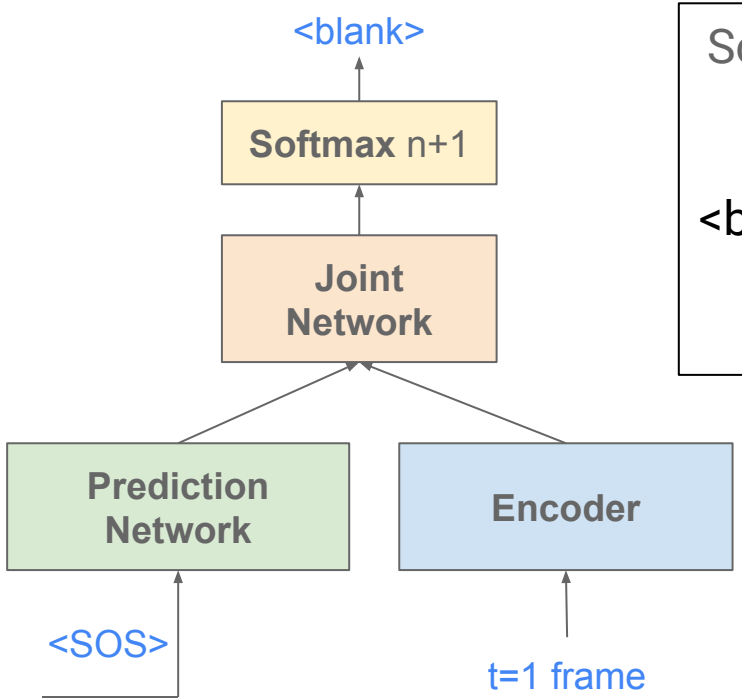
Recurrent Neural Network Transducer (RNN-T)



RNN-T [Graves, 2012] augments CTC encoder with a recurrent neural network LM

Recurrent Neural Network Transducer (RNN-T)

Output:

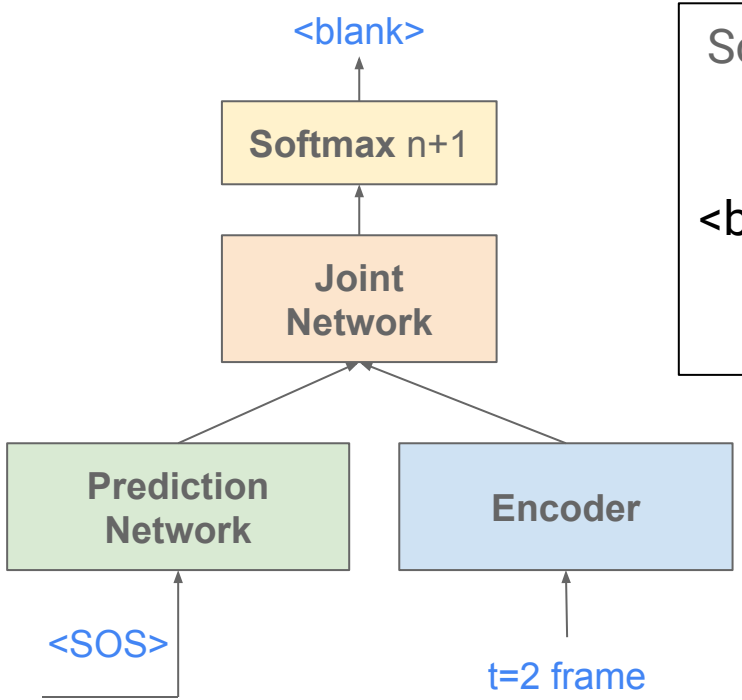


Softmax over n + 1 labels, includes a blank like CTC.

<blank> → advance in Encoder, retain prediction network state

Recurrent Neural Network Transducer (RNN-T)

Output:

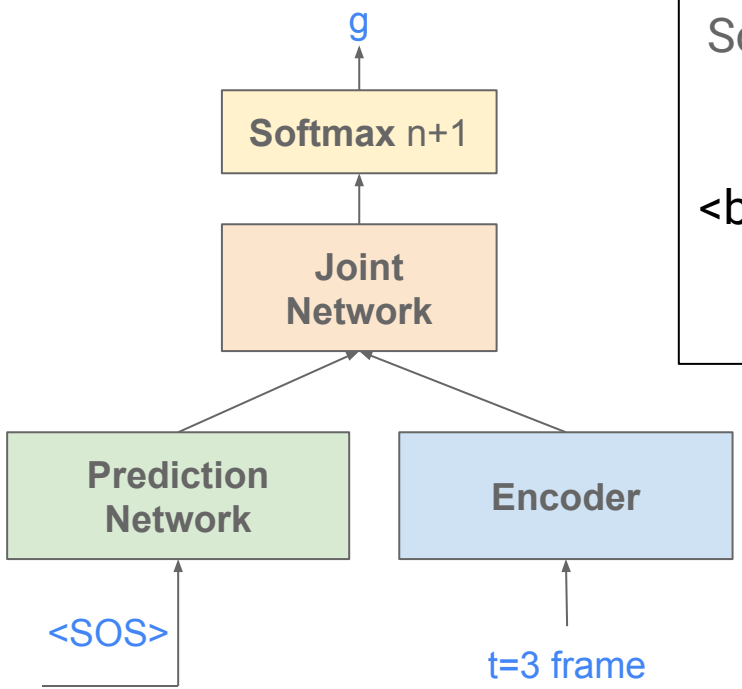


Softmax over n + 1 labels, includes a blank like CTC.

<blank> → advance in Encoder, retain prediction network state

Recurrent Neural Network Transducer (RNN-T)

Output: g

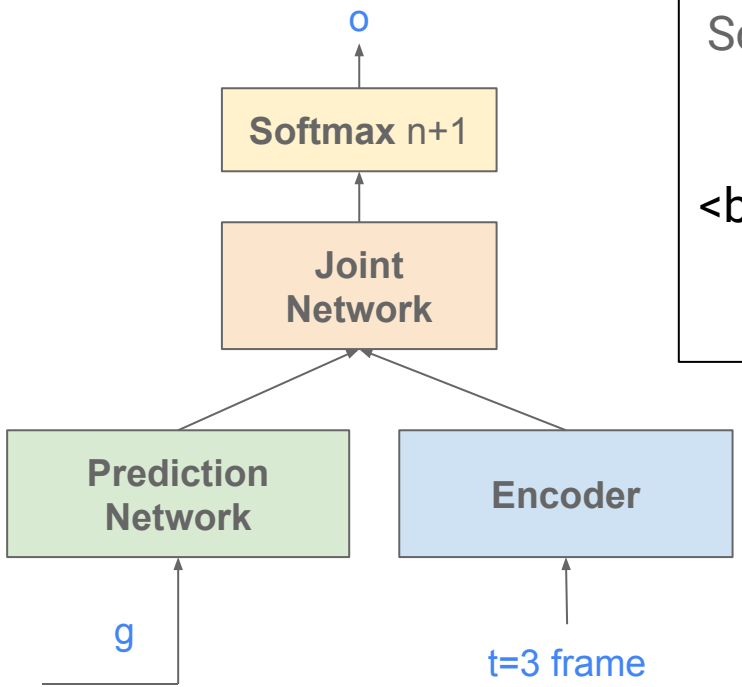


Softmax over n + 1 labels, includes a blank like CTC.

<blank> → advance in Encoder, retain prediction network state

Recurrent Neural Network Transducer (RNN-T)

Output: go

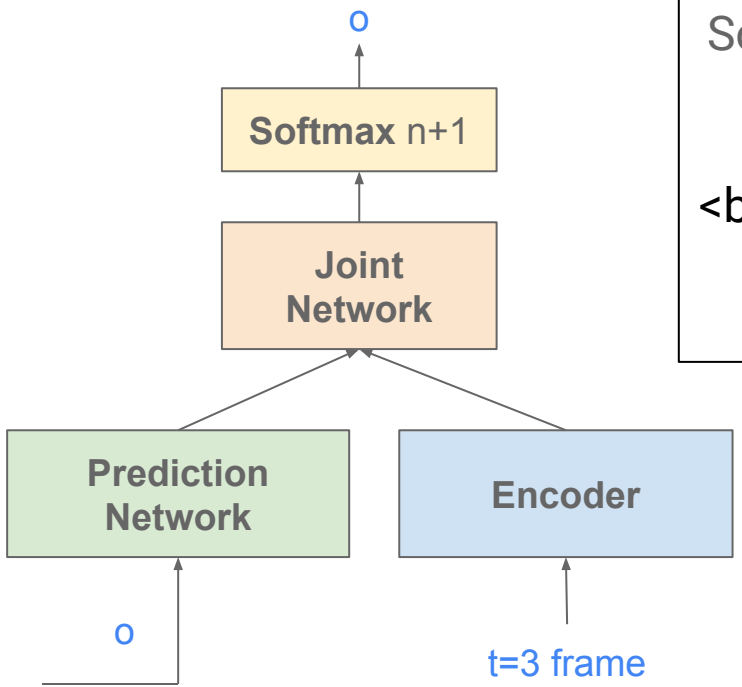


Softmax over n + 1 labels, includes a blank like CTC.

<blank> → advance in Encoder, retain prediction network state

Recurrent Neural Network Transducer (RNN-T)

Output: goo

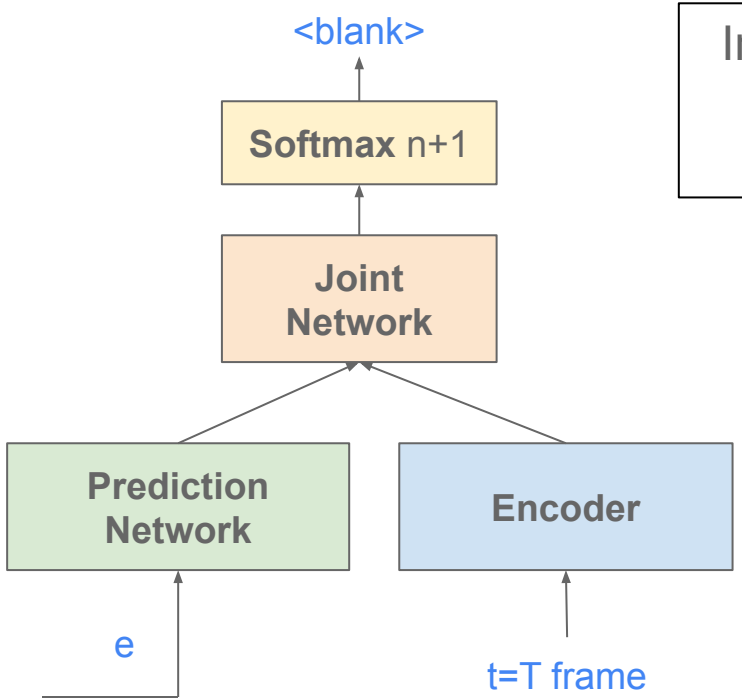


Softmax over n + 1 labels, includes a blank like CTC.

<blank> → advance in Encoder, retain prediction network state

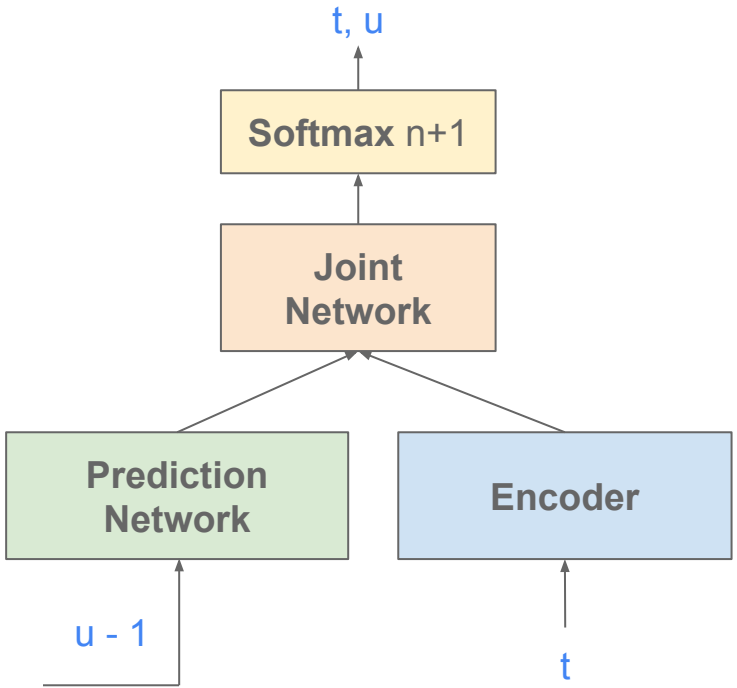
Recurrent Neural Network Transducer (RNN-T)

Output: google

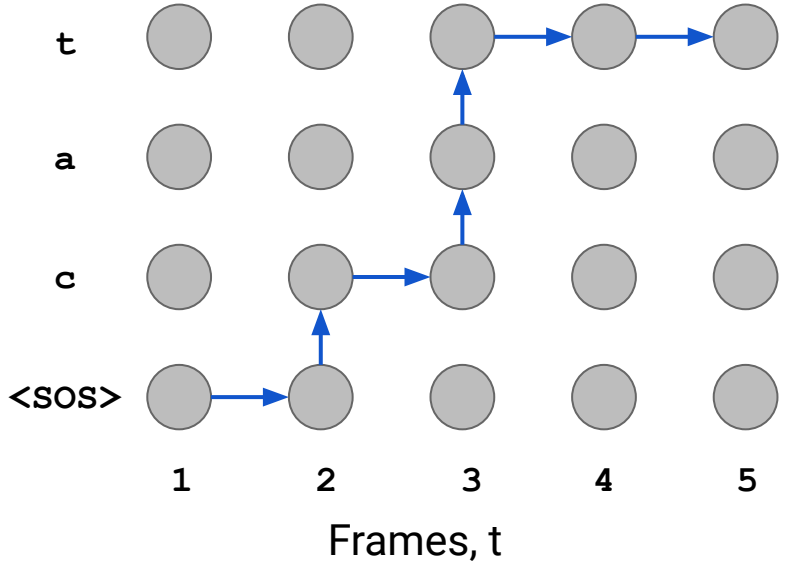


Inference terminates when all input frames have been consumed

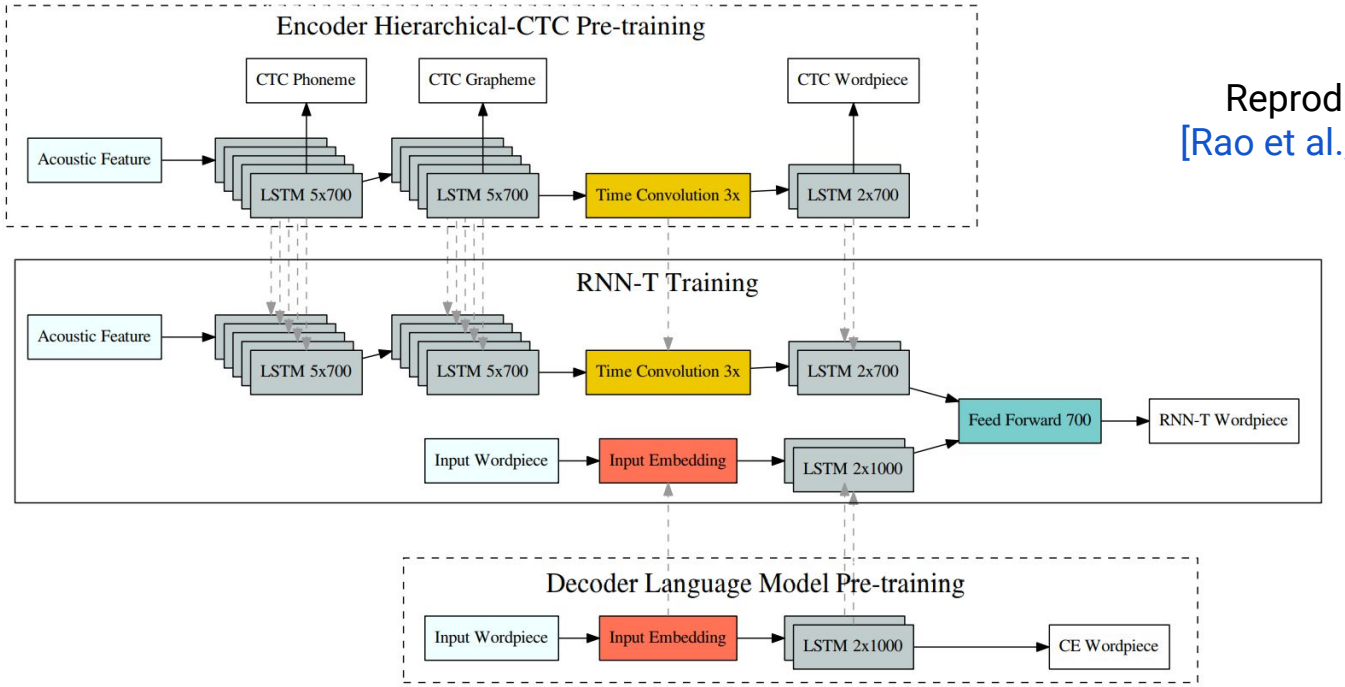
Recurrent Neural Network Transducer (RNN-T)



During training feed the true label sequence to the LM.
Given a target sequence of length U and T acoustic frames we generate UxT softmax



RNN-T: Case Study on ~18,000 hour Google Data



Reproduced from
[Rao et al., 2017] ASRU

RNN-T components can be initialized separately from (hierarchical) CTC-trained AM, and recurrent LM. Initialization generally improves performance.

RNN-T: Case Study on ~18,000 hour Google Data

Units	Layers Encoder	Layers Decoder	Pre-trained Encoder	Pre-trained Decoder	Training Data Used Acoustic	Training Data Used Pronunciation	Training Data Used Text	Params	WER(%) VS	WER(%) IME
RNN-T										
Graphemes	5x700	2x700	no	no	yes	no	no	21M	13.9	8.4
Graphemes	5x700	2x700	yes	no	yes	no	no	21M	13.2	8.0
Graphemes	8x700	2x700	yes	no	yes	no	no	33M	12.0	6.9
Graphemes	8x700	2x700	yes	no	yes	yes	no	33M	11.4	6.8
Graphemes	8x700	2x700	yes	yes	yes	yes	yes	33M	10.8	6.4
Wordpieces-1k	12x700	2x700	yes	yes	yes	yes	yes	55M	9.9	6.0
Wordpieces-10k	12x700	2x700	yes	yes	yes	yes	yes	66M	9.1	5.3
Wordpieces-30k	12x700	2x1000	yes	yes	yes	yes	yes	96M	8.5	5.2
Baseline										
-	-	-	-	-	yes	yes	yes	120.2M	8.3	5.4

Initializing the “encoder” (i.e., acoustic model) helps improve performance by ~5%.

RNN-T: Case Study on ~18,000 hour Google Data

Units	Layers Encoder	Layers Decoder	Pre-trained Encoder	Pre-trained Decoder	Training Data Used Acoustic	Training Data Used Pronunciation	Training Data Used Text	Params	WER(%) VS	WER(%) IME
RNN-T										
Graphemes	5x700	2x700	no	no	yes	no	no	21M	13.9	8.4
Graphemes	5x700	2x700	yes	no	yes	no	no	21M	13.2	8.0
Graphemes	8x700	2x700	yes	no	yes	no	no	33M	12.0	6.9
Graphemes	8x700	2x700	yes	no	yes	yes	no	33M	11.4	6.8
Graphemes	8x700	2x700	yes	yes	yes	yes	yes	33M	10.8	6.4
Wordpieces-1k	12x700	2x700	yes	yes	yes	yes	yes	55M	9.9	6.0
Wordpieces-10k	12x700	2x700	yes	yes	yes	yes	yes	66M	9.1	5.3
Wordpieces-30k	12x700	2x1000	yes	yes	yes	yes	yes	96M	8.5	5.2
Baseline										
-	-	-	-	-	yes	yes	yes	120.2M	8.3	5.4

Initializing the “decoder” (i.e., prediction network, language model) helps improve performance by ~5%.

RNN-T: Case Study on ~18,000 hour Google Data

Units	Layers Encoder	Layers Decoder	Pre-trained Encoder	Pre-trained Decoder	Training Data Used Acoustic	Training Data Used Pronunciation	Training Data Used Text	Params	WER(%) VS	WER(%) IME
RNN-T										
Graphemes	5x700	2x700	no	no	yes	no	no	21M	13.9	8.4
Graphemes	5x700	2x700	yes	no	yes	no	no	21M	13.2	8.0
Graphemes	8x700	2x700	yes	no	yes	no	no	33M	12.0	6.9
Graphemes	8x700	2x700	yes	no	yes	yes	no	33M	11.4	6.8
Graphemes	8x700	2x700	yes	yes	yes	yes	yes	33M	10.8	6.4
Wordpieces-1k	12x700	2x700	yes	yes	yes	yes	yes	55M	9.9	6.0
Wordpieces-10k	12x700	2x700	yes	yes	yes	yes	yes	66M	9.1	5.3
Wordpieces-30k	12x700	2x1000	yes	yes	yes	yes	yes	96M	8.5	5.2
Baseline										
-	-	-	-	-	yes	yes	yes	120.2M	8.3	5.4

The RNN-T model with ~96M parameters can match the performance of a conventional sequence-trained CD-phone based CTC model with a large first pass LM

Towards On-Device Speech Recognition

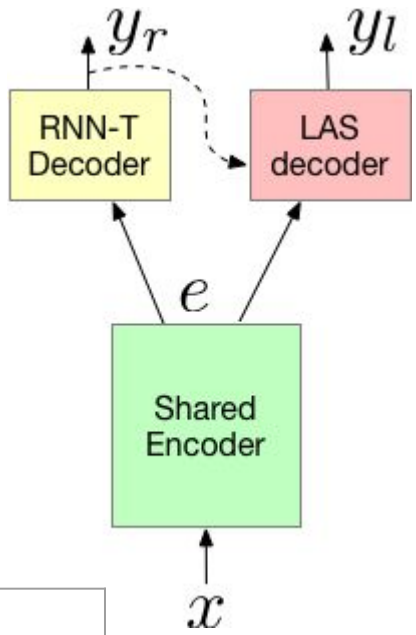
- RNN-T is a great application for on-device speech recognition
  - Fraction the size of the our best server-size conventional model (80GB)
  - Much better performance and RTF compared to on-device conventional CTC model
- Currently used for powering speech input in Gboard
  - Demo
  - Full paper [He et al., 2018]

Model	Size	VS WER	Dictation WER	RT90
On-Device RNN-T	**120MB**	**7.3%**	**4.2%**	**0.51**
On-Device CTC	130MB	9.2%	5.4%	0.86

Real Time (RT) Factor (processing time divided by audio duration) is measured on a Google Pixel Phone.
RT90: Real time factor at 90 percentile. Lower is better.

Towards On-Device Speech Recognition

- Can further include LAS to rescore RNN-T hypotheses, while maintaining streaming solution
- LAS Rescoring gives > 15% WERR over RNN-T
- Comparison to conventional server model is neutral
- Full paper [Sainath et al., to appear in Interspeech 2018]



Models	Short Utterances	Long Utterances
Baseline RNN-T	6.9	4.5
LAS Rescoring	**5.7**	**3.5**

Side-by-Side Comparing Conventional Server Model to End-To-End Model				
Changed (%)	Win	Loss	Neutral	p-Value
13.2	48	61	391	10.0%-20.0%

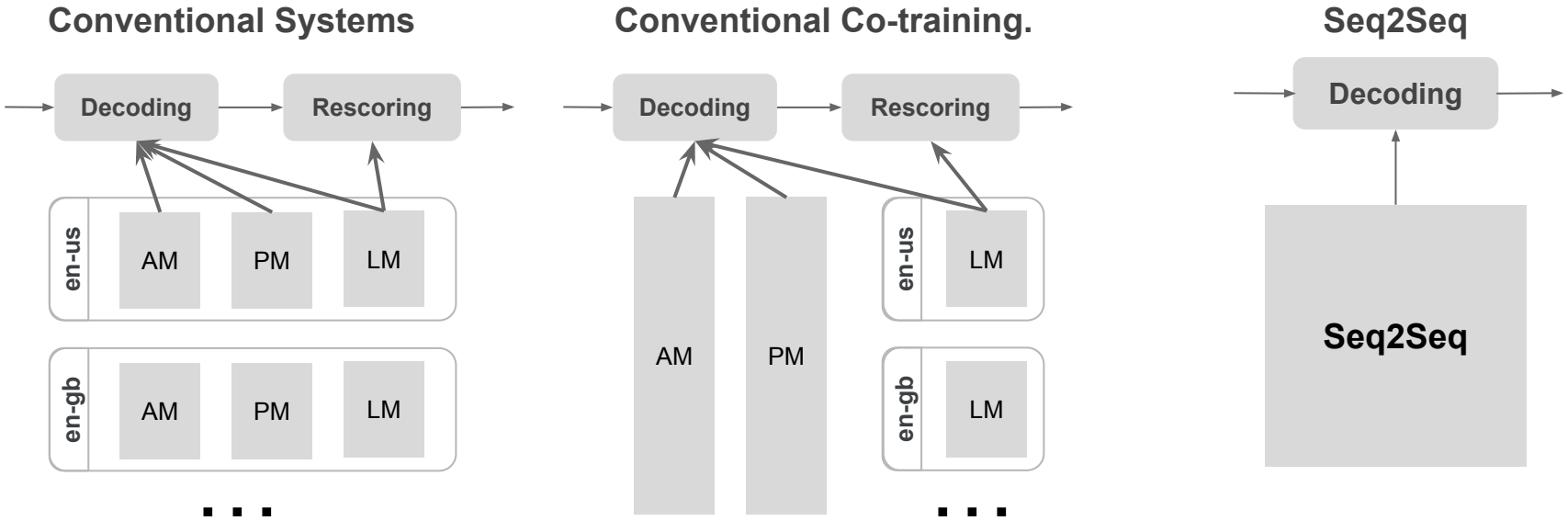
SXS Analysis

Prod Correct	LAS Incorrect	Prod Incorrect	LAS Correct
the podcast **serial**	the podcast **cereal**	**ideal image.com**	**idealimage.com**
who played horatio **caine** on csi miami	who played horatio **kane** on csi miami	petting zoo **chicagoland** area	petting zoo **chicago land** area
Hawthorn amc theaters	**Hawthorne** amc theaters	nick **mangold** current team	nick **mangold's** current team
18ft duckworth advantage tiller review	**18-ft** duckworth advantage tiller review	**gena rowlands**	**jenna rowlands**

Prod does much better on proper nouns
Many E2E wins are minor prod mistakes

Future Directions: Multi-lingual End-to-End

E2E multi-dialect ASR



In conventional systems, languages/dialects, are handled with **individual AMs, PMs and LMs**. Upscaling is becoming challenging.

A single model for all.

Multi-Dialect LAS

- Modeling Simplicity
- Data Sharing
  - among dialects and model components
- Joint Optimization
- Infrastructure Simplification
  - a single model for all

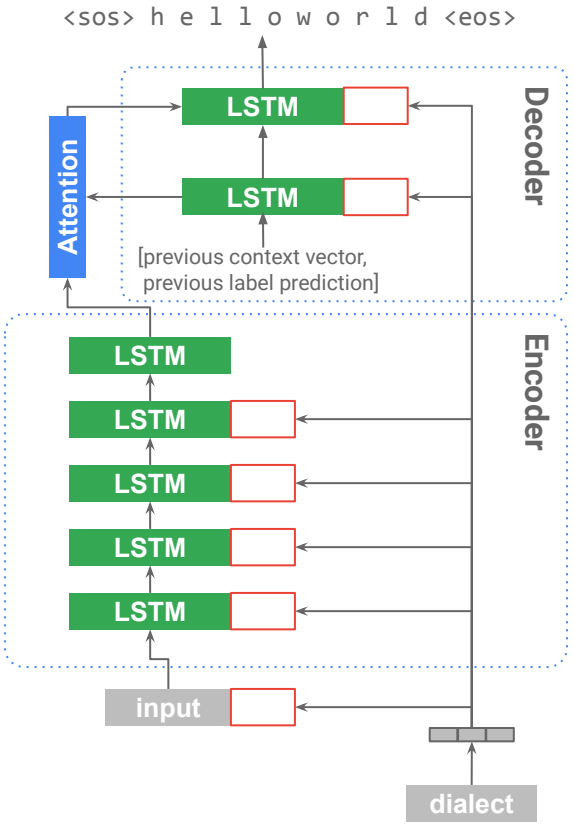
Table: Resources required for building each system.

Conventional	Seq2Seq
data phoneme lexicon text normalization LM × N	data

Dialect as Input Features

- Passing the dialect information as additional features

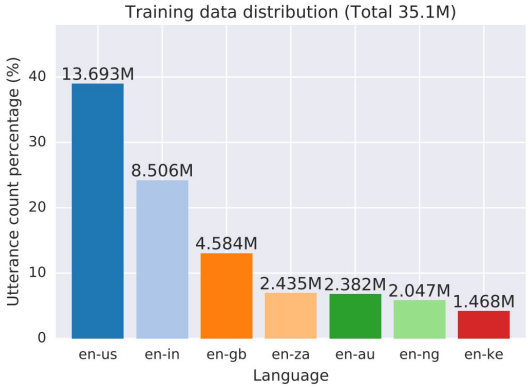
components		variations
encoders	→	acoustic
decoders	→	lexicon and language



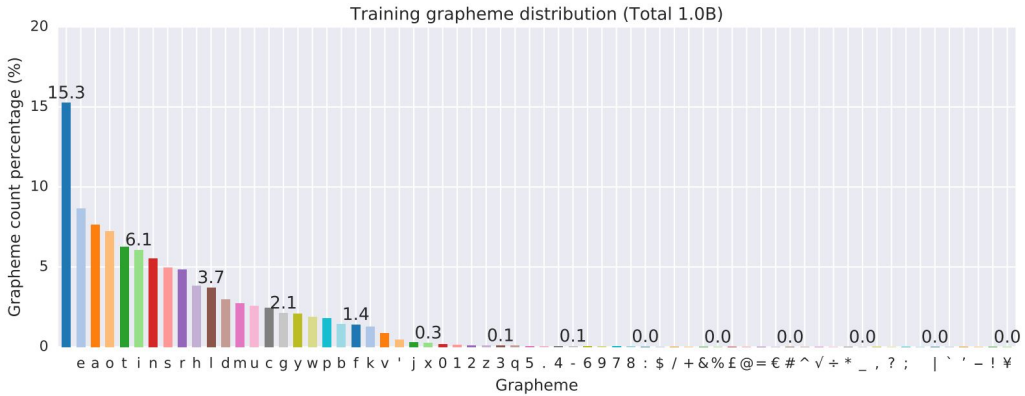
[B. Li, ICASSP 2018]

Task

- **7 English dialects:** US (America), IN (India), GB (Britain), ZA (South Africa), AU (Australia), NG (Nigeria & Ghana), KE (Kenya)



★ **unbalanced** dialect data



★ **unbalanced** target classes

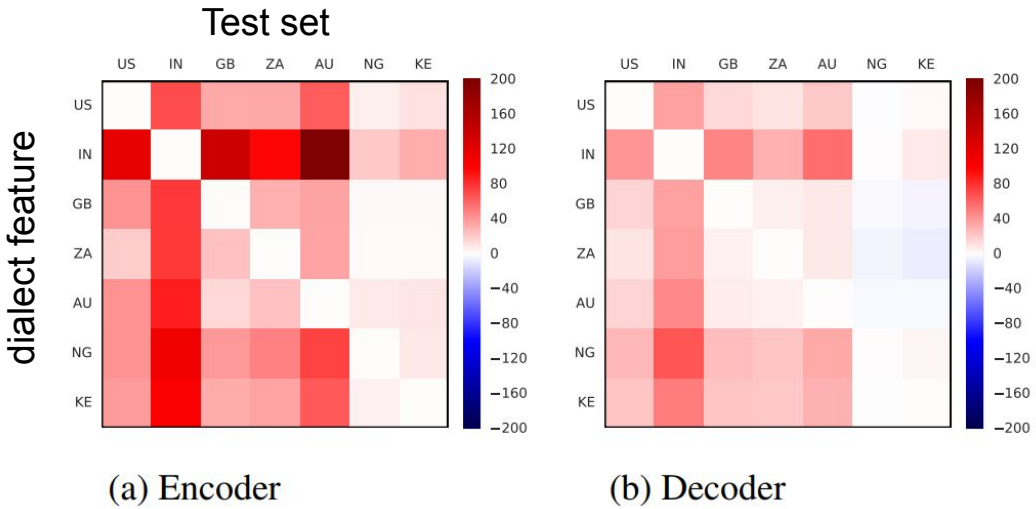
LAS With Dialect as Input Features

Dialect	US	IN	GB	ZA	AU	NG	KE
Baseline (dialect-dep.)	9.7	16.2	12.7	11.0	12.1	33.4	19.0
encoder	9.6	16.4	11.8	10.6	10.7	31.6	18.1
decoder	9.4	16.2	**11.3**	10.8	10.9	32.8	18.0
both	**9.1**	**15.7**	11.5	**10.0**	**10.1**	**31.3**	**17.4**

★ feeding dialect to **both encoder and decoder** gives the largest gains

LAS With Dialect as Input Features

Feeding different dialect vectors (rows) on different test sets (columns).



★ **encoder** is more sensitive to wrong dialects → large acoustic variations

★ for **low-resource** dialects (NG, KE), the model **learns to ignore** the dialect information

LAS With Dialect as Input Features

- The dialect vector does **both AM and LM adaptation**

*Table: The number of **color**/**colour** occurrences in hypotheses on the **en-gb** test data.*

dialect vector	encoder	decoder	color (US)	colour (GB)
❌	❌	❌	1	22
<en-gb>: [0, 1, 0, 0, 0, 0, 0]	✓	❌	19	4
<en-gb>: [0, 1, 0, 0, 0, 0, 0]	❌	✓	0	25
<en-us>: [1, 0, 0, 0, 0, 0, 0]	❌	✓	24	0

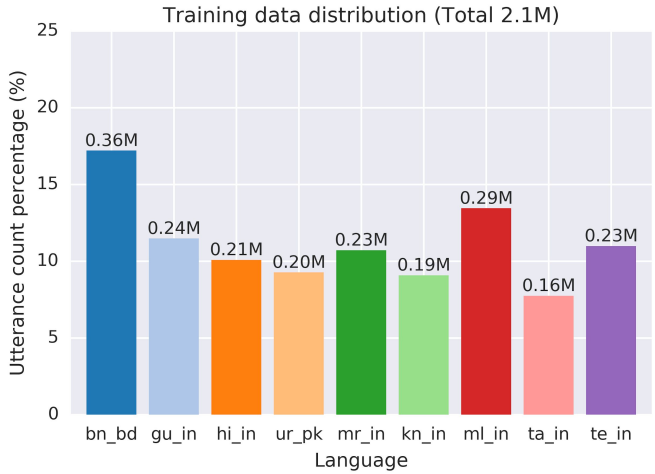
★ dialect vector helps **encoder** to **normalize accent variations**

★ dialect vector helps **decoder** to **learn dialect-specific lexicons**

IndicX - Task

[S. Toshniwal, ICASSP 2018]

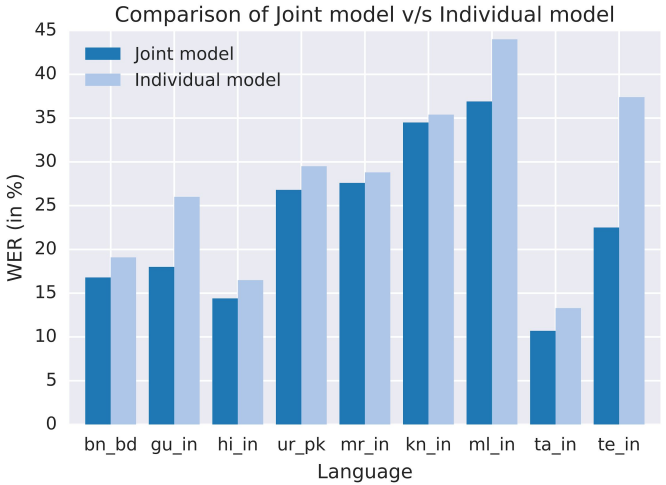
- **9 Indian languages:** bn_bd, gu_in, hi_in, ur_pk, mr_in, kn_in, ml_in, ta_in, te_in
- **large script variations**



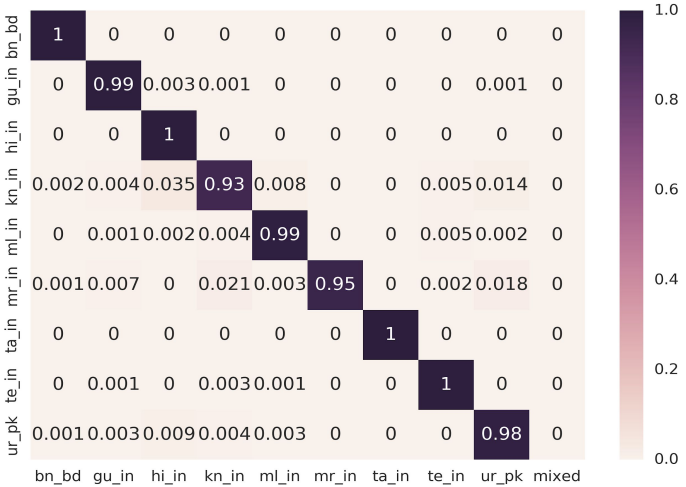
- Bengali (bn_bd) - আমার বাবা ওদেরকে বলতেন
- Gujarati (gu_in) - હું ઘરની અંદર ન મરું અને બહાર પણ ન મરું
- Hindi (hi_in) - पहले वीडियोग्राफी होगी
- Kannada (kn_in) - ಮುಖದ ಮಧ್ಯದಲ್ಲಿ ಪಿಷ್ಟ
- Malayalam (ml_in) - എന്നിട്ടും അവരുടെ വാക്കുകളിലൂടെ അവരെ
- Marathi (mr_in) - श्रीकृष्णाच्या गोकुळातल्या
- Tamil (ta_in) - இது ஒரு நகராட்சியாகும்
- Telugu (te_in) - ఈ పేజీని 'తర్జుమా' చేయకముందు ఇవికీలో పెడదామా
- Urdu (ur_pk) - مصنف کا کلام جو ڑھوگر مےالرحبدع خ ےش

[1] S. Toshniwal, T.N. Sainath, R.J. Weiss, B. Li et.al Multilingual speech recognition with a single end-to-end model, *ICASSP 2018*

IndicX - Co-training



★ co-trained model is consistently better



★ co-trained model chooses the right script

★ tested multitask learning (LID and ASR), not helpful

★ the model cannot do code switching, faithful to one language

Google [1] S. Toshniwal, T.N. Sainath, R.J. Weiss, B. Li et.al Multilingual speech recognition with a single end-to-end model, *ICASSP 2018*

Summary and Challenges

Attention

- Attention-based end-to-end model achieves **state-of-the-art** performance on Google's Voice Search but is **not streaming**

Online Models

- Recurrent Neural Network Transducer (RNNT) is **streaming** and can be used for **on-device ASR**

Multi-lingual E2E

- E2E greatly simplifies multi-lingual/dialect E2E speech recognition

Open Questions: Handling the Long Tail

- E2E models do very poorly on rare words/proper nouns:
  - Very little occurence in training data
  - E2E model is trained on audio-text pairs, fraction of data compared to a conventional text-only LM
- This problem is exacerbated with contextual biasing

Lingvo (tensorflow/lingvo)

A toolkit suited to build neural networks, particularly sequence models.

Machine Translation

Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation.

... ...

Speech Recognition

State-of-the-art Speech Recognition With Sequence-to-Sequence Models.

... ...

Speech Synthesis

Hierarchical Generative Modeling for Controllable Speech Synthesis.

... ...

Language Understanding

Semi-Supervised Learning for Information Extraction from Dialogue.

... ...

Source: https://github.com/tensorflow/lingvo

Thank You

References

[Audhkhasi et al., 2017] K. Audhkhasi, B. Ramabhadran, G. Saon, M. Picheny, D. Nahamoo “Direct Acoustics-to-Word Models for English Conversational Speech Recognition,” Proc. of Interspeech, 2017.

[Bahdanau et al., 2017] D. Bahdanau, P. Brakel, K. Xu, A. Goyal, R. Lowe, J. Pineau, A. Courville, Y. Bengio, “An Actor-Critic Algorithm for Sequence Prediction,” Proc. of ICLR, 2017.

[Battenberg et al., 2017] E. Battenberg, J. Chen, R. Child, A. Coates, Y. Gaur, Y. Li, H. Liu, S. Satheesh, D. Seetapun, A. Sriram, Z. Zhu, “Exploring Neural Transducers For End-to-End Speech Recognition,” Proc. of ASRU, 2017.

[Chan et al., 2015] W. Chan, N. Jaitly, Q. V. Le, and O. Vinyals, “Listen, Attend and Spell,” CoRR, vol. abs/1508.01211, 2015.

[Chang et al., 2017] S-Y. Chang, B. Li, T. N. Sainath, G. Simko, C. Parada, “Endpoint Detection using Grid Long Short-Term Memory Networks for Streaming Speech Recognition,” Proc. of Interspeech, 2017.

[Chang et al., 2018] S-Y. Chang, B. Li, G. Simko, T. N. Sainath, A. Tripathi, A. van den Oord, O. Vinyals, “Temporal Modeling Using Dilated Convolution and Gating for Voice-Activity-Detection,” Proc. of ICASSP, 2018.

[Chiu and Raffel, 2017] C.-C. Chiu, C. Raffel, “Monotonic Chunkwise Alignments,” Proc. of ICLR, 2017.

[Chiu et al., 2018] C.-C. Chiu, T. N. Sainath, Y. Wu, R. Prabhavalkar, P. Nguyen, Z. Chen, A. Kannan, R. J. Weiss, K. Rao, E. Gonina, N. Jaitly, B. Li, J. Chorowski, M. Bacchiani, “State-of-the-art Speech Recognition With Sequence-to-Sequence Models,” Proc. of ICASSP, 2018.

[Chorowski et al., 2015] J. K. Chorowski, D. Bahdanau, D. Serdyuk, K. Cho, and Y. Bengio, “Attention-Based Models for Speech Recognition,” in Proc. of NIPS, 2015.

[Graves et al., 2006] A. Graves, S. Fernandez, F. Gomez, J. Schmidhuber, “Connectionist Temporal Classification: Labelling Unsegmented Sequence Data with Recurrent Neural Networks,” Proc. of ICML, 2006.

[Graves, 2012] A. Graves, “Sequence Transduction with Recurrent Neural Networks,” Proc. of ICML Representation Learning Workshop, 2012.

References

[Graves et al., 2013] A. Graves, A. Mohamed, and G. Hinton, “Speech Recognition with Deep Neural Networks,” in Proc. ICASSP, 2013.

[Gulcehre et al., 2015] C. Gulcehre, O. Firat, K. Xu, K. Cho, L. Barrault, H.-C. Lin, F. Bougares, H. Schwenk, Y. Bengio, “On Using Monolingual Corpora in Neural Machine Translation”, CoRR, vol. abs/1503.03535, 2015.

[Hannun et al., 2014] A. Hannun, A. Maas, D. Jurafsky, A. Ng, “First-Pass Large Vocabulary Continuous Speech Recognition using Bi-Directional Recurrent DNNs,” CoRR, vol. abs/1408.2873, 2014.

[He et al., 2017] Y. He, R. Prabhavalkar, K. Rao, W. Li, A. Bakhtin and I. McGraw, "Streaming small-footprint keyword spotting using sequence-to-sequence models," Proc. of ASRU, 2017.

[He et al., 2018] Y. He, T. N. Sainath, R. Prabhavalkar, I. McGraw, R. Alvarez, D. Zhao, D. Rybach, A. Kannan, Y. Wu, R. Pang, Q. Liang, D. Bhatia, Y. Shangguan, B. Li, G. Pundak, K. C. Sim, T. Bagby, S-Y. Chang, K. Rao, A. Gruenstein, “Streaming End-to-end Speech Recognition For Mobile Devices,” CoRR, vol. abs/1811.06621, 2018.

[Jaitly et al., 2016] N. Jaitly, D. Sussillo, Q. V. Le, O. Vinyals, I. Sutskever, S. Bengio, “An Online Sequence-to-Sequence Model Using Partial Conditioning,” Proc. of NIPS, 2016.

[Kannan et al., 2018] A.Kannan, Y. Wu, P. Nguyen, T. N. Sainath, Z. Chen, R. Prabhavalkar, "An analysis of incorporating an external language model into a sequence-to-sequence model," Proc. of ICASSP, 2018.

[Kim and Rush, 2016] Y. Kim and A. M. Rush, “Sequence-level Knowledge Distillation,” Proc. of EMNLP, 2016.

[Kim et al., 2017] S. Kim, T. Hori and S. Watanabe, “Joint CTC-attention based End-to-End Speech Recognition using Multi-Task Learning,” Proc. of ICASSP, 2017.

[Kingsbury, 2009] B. Kingsbury, “Lattice-based optimization of sequence classification criteria for neural-network acoustic modeling,” Proc. of ICASSP, 2009.

[Li et al., 2018] B. Li, T. N. Sainath, K. C. Sim, M. Bacchiani, E. Weinstein, P. Nguyen, Z. Chen, Y. Wu, K. Rao, “Multi-Dialect Speech Recognition With A Single Sequence-To-Sequence Model,” Proc. of ICASSP, 2018.

References

[Maas et al., 2015] A. Maas, Z. Xie, D. Jurafsky, A. Ng, “Lexicon-Free Conversational Speech Recognition with Neural Networks," Proc. of NAACL-HLT, 2015.

[McGraw et al., 2016] I. McGraw, R. Prabhavalkar, R. Alvarez, M. G. Arenas, K. Rao, D. Rybach, O. Alsharif, H. Sak, A. Gruenstein, F. Beaufays, C. Parada, “Personalized speech recognition on mobile devices”, Proc. of ICASSP, 2016

[Rabiner, 1989] L. R. Rabiner, “A tutorial on hidden markov models and selected applications in speech recognition,” Proc. of IEEE, 1989.

[Prabhavalkar et al., 2017] R. Prabhavalkar, K. Rao, T. N. Sainath, B. Li, L. Johnson, N. Jaitly, “A Comparison of Sequence-to-Sequence Models for Speech Recognition,” Proc. of Interspeech, 2017.

[Prabhavalkar et al., 2018] R. Prabhavalkar, T. N. Sainath, Y. Wu, P. Nguyen, Z. Chen, C.-C. Chiu, A. Kannan, “Minimum Word Error Rate Training for Attention-based Sequence-to-Sequence Models,” Proc. of ICASSP, 2018.

[Povey, 2003] D. Povey, “Discriminative Training for Large Vocabulary Speech Recognition”, Ph.D. thesis, Cambridge University Engineering Department, 2003.

[Pundak et al., 2018] G. Pundak, T. N. Sainath, R. Prabhavalkar, A. Kannan, D. Zhao, “Deep context: end-to-end contextual speech recognition,” Proc. of SLT, 2018.

[Rao et al., 2017] K. Rao, H. Sak, R. Prabhavalkar, “Exploring Architectures, Data and Units For Streaming End-to-End Speech Recognition with RNN-Transducer”, Proc. of ASRU, 2017.

[Ranzato et al., 2016] M. Ranzato, S. Chopra, M. Auli, and W. Zaremba, “Sequence level training with recurrent neural networks,” Proc. of ICLR, 2016.

[Sainath et al., 2018] T. N. Sainath, C.-C. Chiu, R. Prabhavalkar, A. Kannan, Y. Wu, P. Nguyen, Z. Chen, “Improving the Performance of Online Neural Transducer Models,” Proc. of ICASSP, 2018.

[Sak et al., 2015] Has¸im Sak, Andrew Senior, Kanishka Rao, Francoise Beaufays, “Fast and Accurate Recurrent Neural Network Acoustic Models for Speech Recognition,” Proc. of Interspeech, 2015.

References

[Sak et al., 2017] H. Sak, M. Shannon, K. Rao, and F. Beaufays, “Recurrent neural aligner: An encoder-decoder neural network model for sequence to sequence mapping,” in Proc. of Interspeech, 2017.

[Schuster & Nakajima, 2012] M. Schuster and K. Nakajima, “Japanese and Korean Voice Search,” Proc. of ICASSP, 2012.

[Shannon, 2017] M. Shannon, “Optimizing expected word error rate via sampling for speech recognition,” in Proc. of Interspeech, 2017.

[Sim et al., 2017] K. Sim, A. Narayanan, T. Bagby, T. N. Sainath, and M. Bacchiani, “Improving the Efficiency of Forward-Backward Algorithm using Batched Computation in TensorFlow,” Proc. of ASRU, 2017.

[Sriram et al., 2018] A. Sriram, H. Jun, S. Satheesh, A. Coates, “Cold Fusion: Training Seq2Seq Models Together with Language Models,” Proc. of ICLR, 2018.

[Stolcke et al., 1997] A. Stolcke, Y. Konig, M. Weintraub, “Explicit word error minimization in N-best list rescoring,” Proc. of Eurospeech, 1997.

[Su et al., 2013] H. Su, G. Li, D. Yu, and F. Seide, “Error back propagation for sequence training of context-dependent deep networks for conversational speech transcription,” Proc. of ICASSP, 2013.

[Szegedy et al., 2016] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, “Rethinking the inception architecture for computer vision,” Proc. of CVPR, 2016.

[Toshniwal et al., 2018] S. Toshniwal, T. N. Sainath, R. J. Weiss, B. Li, P. Moreno, E. Weinstein, K. Rao, “Multilingual Speech Recognition With A Single End-To-End Model,” Proc. of ICASSP, 2018.

[Vaswani et al., 2017] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, “Attention Is All You Need,” Proc. of NIPS, 2017.

[Wiseman and Rush, 2016] S. Wiseman and A. M. Rush, “Sequence-to-Sequence Learning as Beam Search Optimization,” Proc. of EMNLP, 2016.

Backup

Language Model

Motivation #1

Reference	LAS model output
What language is built into electrical circuitry of a computer?	what language is built into electrical circuit tree of a computer
Leona Lewis believe	vienna lewis believe
Suns-Timberwolves score	sun's timberwolves score

Some Voice Search errors appear to be fixable with a good language model trained on more text-only data.

Motivation #2

- The LAS model requires audio-text pairs: we have only 15M of these
- Our production LM is trained on billions of words of text-only data
- How can we look at incorporating a larger LM into our LAS model?
- More details can be found in [Kannan et al., 2018]

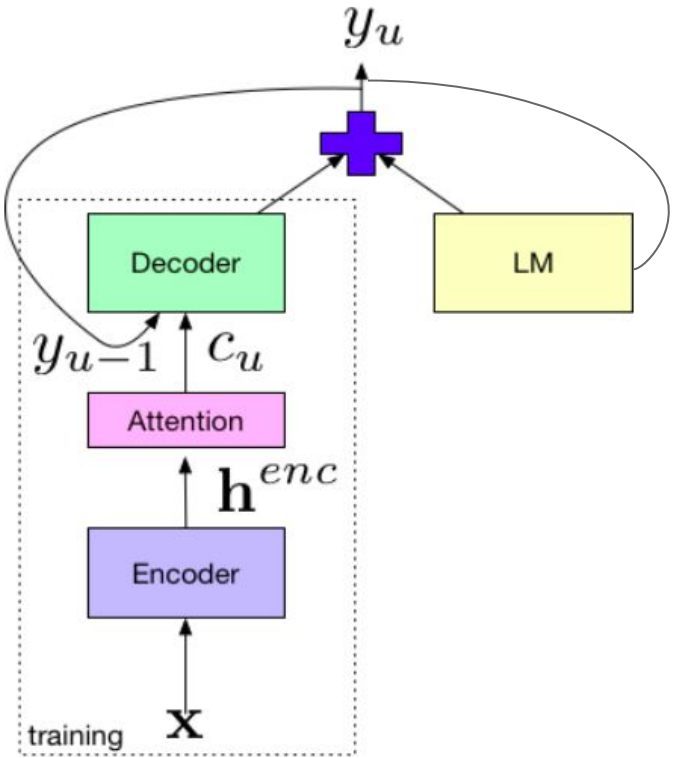
Shallow fusion

- Log-linear interpolation between language model and seq2seq model:

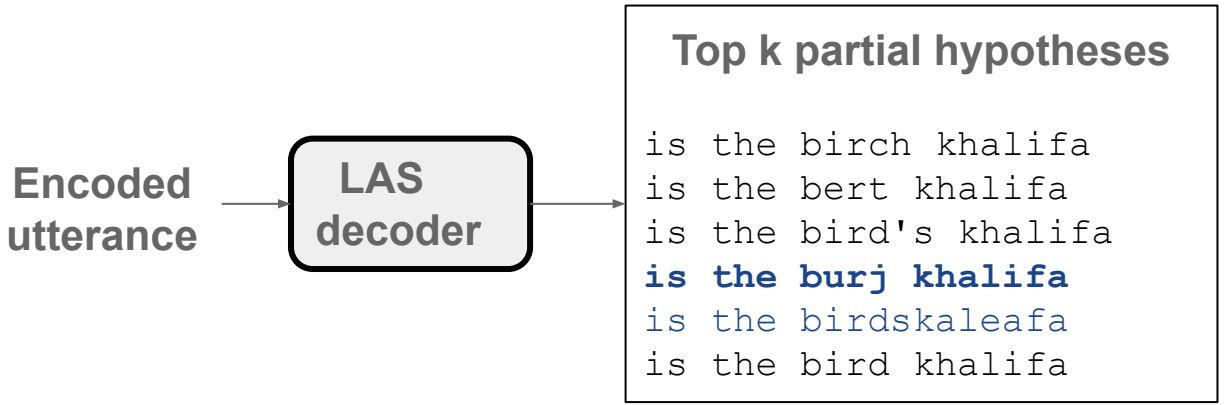
$$\mathbf{y}^* = \arg\max_y \log p(y|x) + \lambda \log p_{LM}(y)$$

- Typically only performed at inference time
- Language model is trained ahead of time and fixed
- LM can be either n-gram (FST) or RNN.
- Analogous to 1st pass rescoring.
- [Chorowski and Jaitly, 2017]. [Kannan et al., 2018].

Shallow fusion

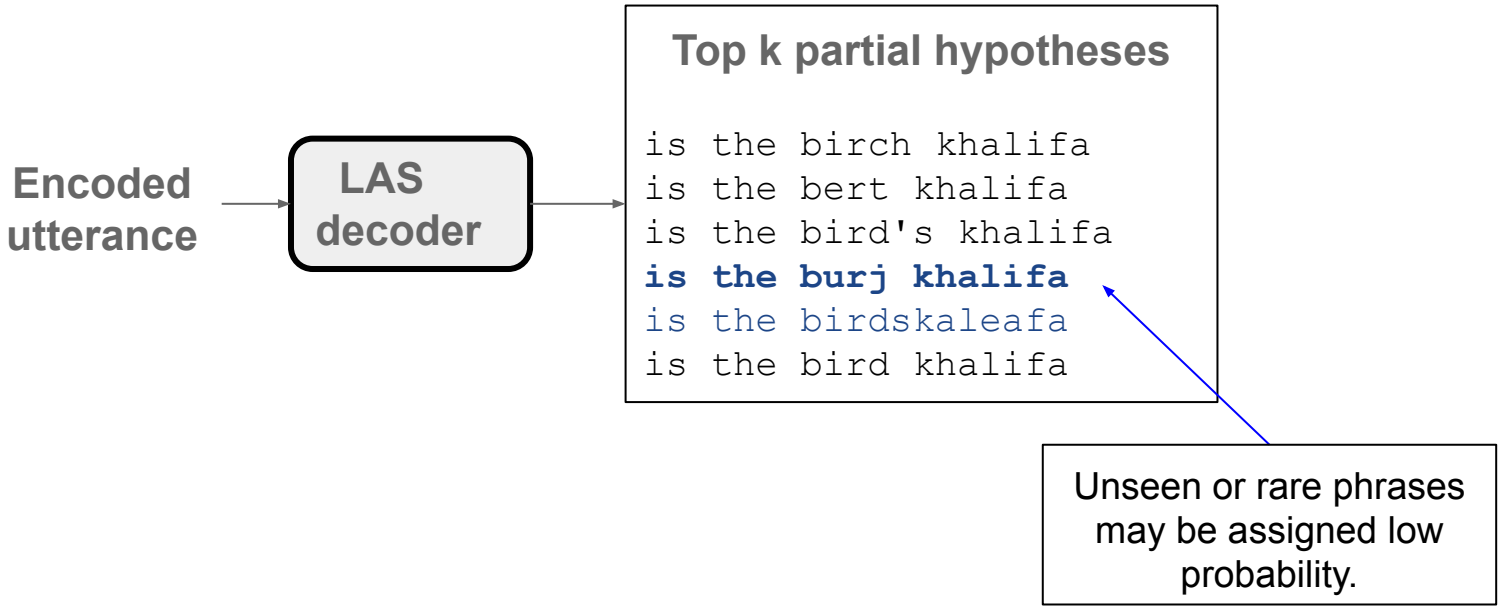


Baseline LAS Model

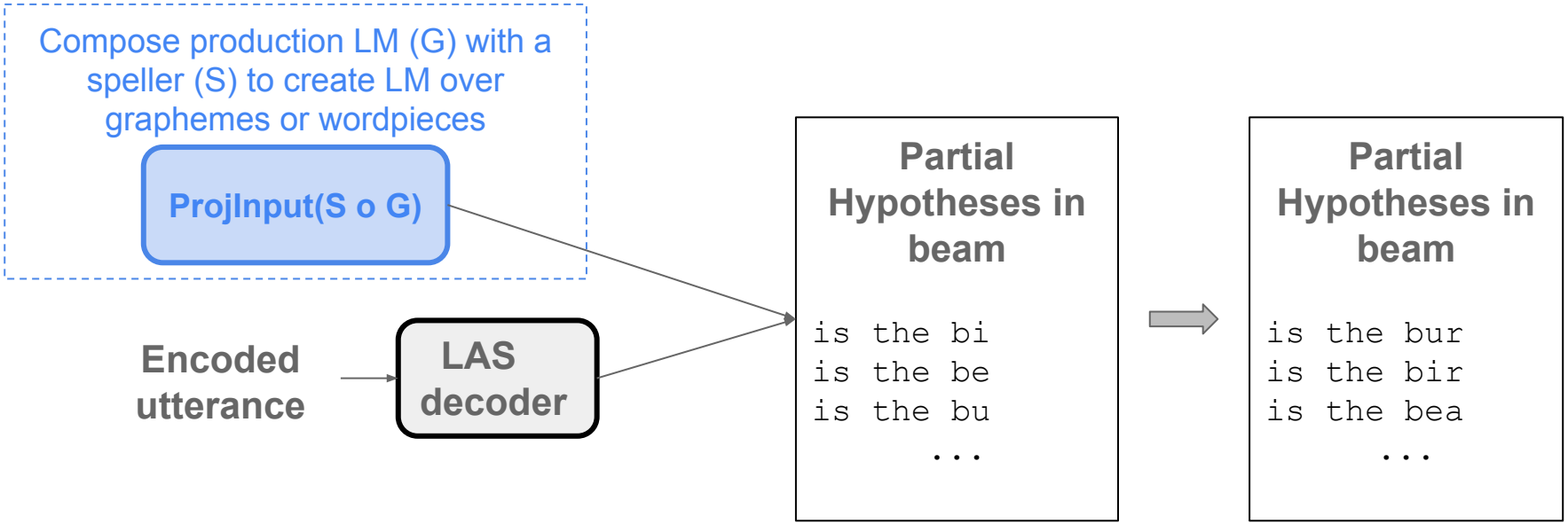


Baseline LAS model relies on LM learned from train data

Baseline LAS Model

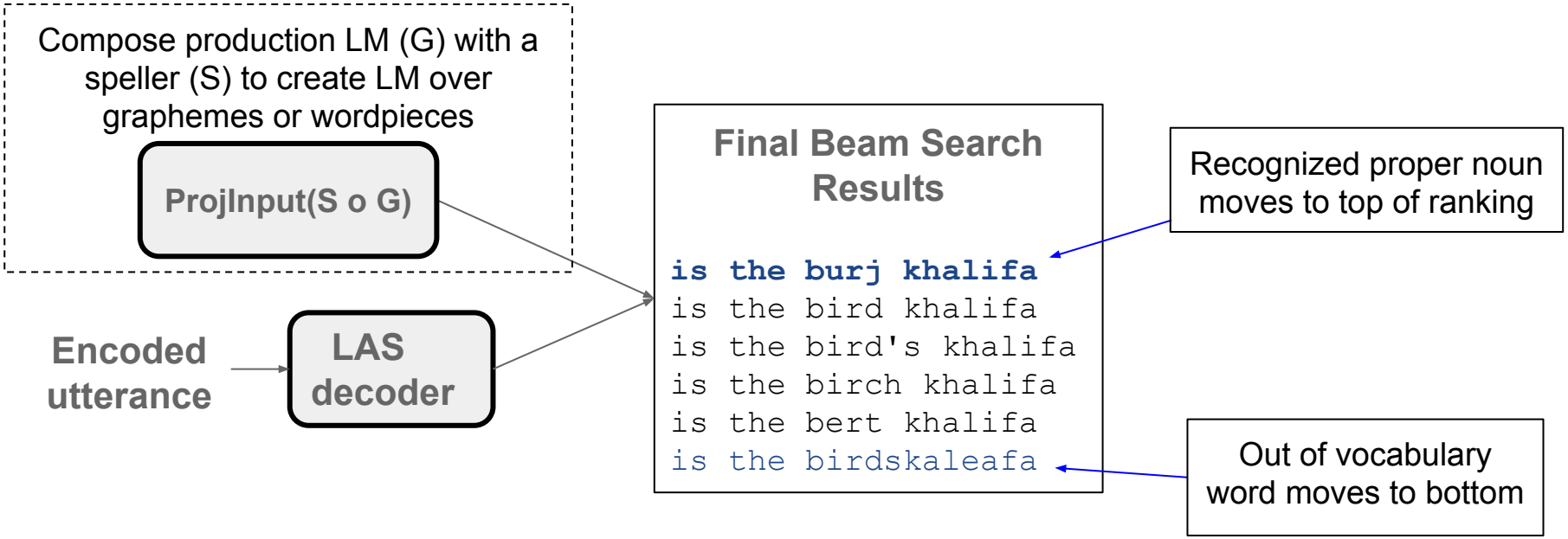


Integration with FST LM in 1st pass



Interpolate model posteriors with LM-score at *each* step of next label prediction

Integration with FST LM in 1st pass



Results with FST LM

System	Dev WER	Test WER	LM Size
Baseline LAS	9.2%	7.7%	0 GB
LAS + FST LM in 1st pass	8.8%	7.4%	2 GB

Decoding with FST 1st pass production LMs into LAS system provides small improvement

Examples of LM wins

	Reference	Top 1 without LM	Top 1 with LM
Rare words	achondroplasia	acondra placia	achondroplasia
Proper nouns	st. isaac jogues mass schedule	st isaac jog's mass schedule	st isaac jogues mass schedule
	what causes high latency on a wi-fi connection?	what causes highlight and sienna wi-fi connection	what causes high latency on a wi-fi connection

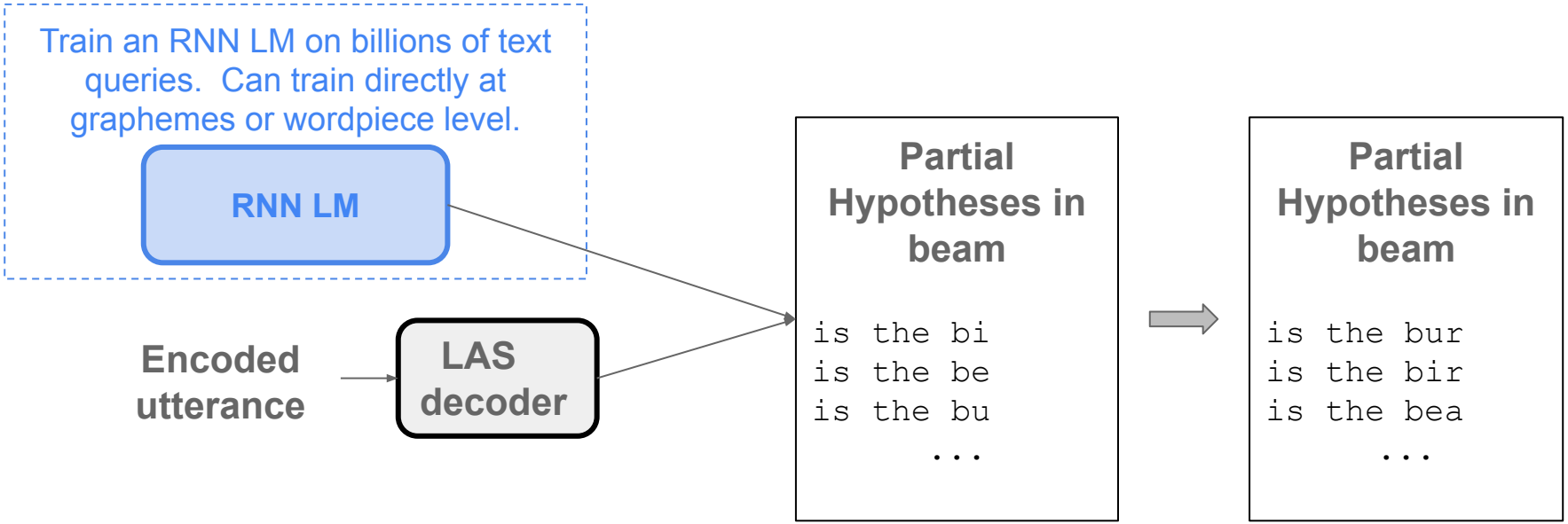
Decoding with LM can correct errors early in decoding.
In examples above, correct hypothesis does not appear in N-best without LM, so would not be possible to correct in second-pass with ProdLM.

Examples of LM losses

	Reference	Top 1 without LM	Top 1 with LM
Out of vocab terms	urgent important unurgent unimportant	urgent important unurgent unimportant	urgent important un urgent unimportant
Websites (specific type of OOV)	mathfunbook.com product of a power property	mathfunbook.com product of a power property	math funbook com product of a power property
Grammatically incorrect language	why you not listening to me tonight	why you not listening to me tonight	why are you not listening to me tonight

LAS model can actually output words it has never seen before.
Decoding with a language model removes this ability, costing about 0.2% absolute WER.

Alternative: integrate with RNN LM in 1st pass



RNN LM can achieve lower perplexity than *n*-gram LM and does not suffer from OOV problem.

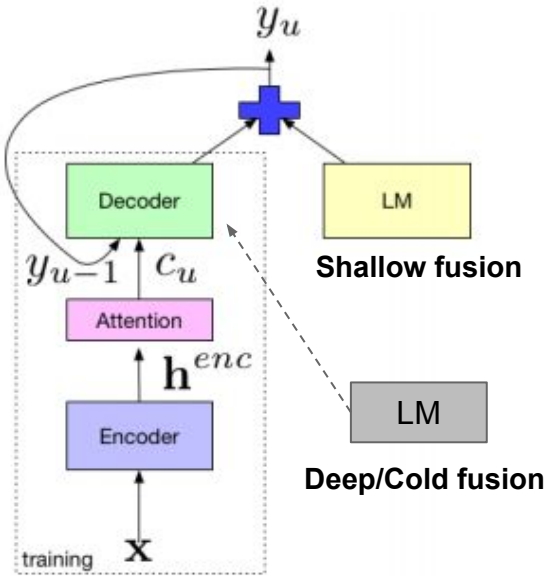
Results with RNN-LM

System	Dev WER	Test WER	LM Size
Baseline LAS	9.2%	7.7%	0 GB
LAS + FST LM in 1st pass	8.8%	7.4%	2 GB
LAS + RNN LM in 1st pass	**8.4%**	**7.0%**	**1 GB**

Decoding with RNN LM provides greater improvement at half the size!

Extending LAS with an LM

- Listen, Attend and Spell [Chan et al., 2015]
- How to incorporate an LM?
  - **Shallow fusion** [Kannan et al., 2018]
    - LM is applied on output
  - **Deep fusion** [Gulcehre et al., 2015]
    - Assumes LM is fixed
  - **Cold fusion** [Sriram et al., 2017]
    - Simple interface between a deep lm and the encoder
    - Allows to swap in task-specific LMs
- In these experiments, fusion is used during the beam search rather than n-best rescoring.



Comparison of Fusion Results

- Shallow Fusion still seems to perform the best
- Full comparison in [Toshniwal, 2018]

System	Voice Search	Dictation
Baseline LAS	5.6	4.0
Shallow Fusion	**5.3**	**3.7**
Deep Fusion	5.5	4.1
Cold Fusion	**5.3**	3.9

Handling Long Tail with Biasing

What is “Biasing”?

“An attempt to adapt the priors baked into the speech models to better model information gained between training and inference (aka context).”

Why Is Biasing Important

- Biasing can improve WER in domains by more than 10% relative

Test Set	WER, No Biasing	WER, Biasing
Contacts	15.0	2.8
Numeric	11.0	4.7
Yes-No-Cancel	18.8	10.4

How To Bias E2E Models

- Two options for biasing
  - Bias externally
  - Biasing within the model
- Paper reference [Pundak et al., 2018]
- In these experiments, we will evaluate on the following test sets
  - Contacts - “call Joe Doe, send a message to Jason Dean”
  - Songs - “play Lady Gaga, play songs from Jason Mraz”
  - Third Party - “text Jeanne, text John”

(1) Biasing - Shallow Fusion

- General equation for shallow fusion during beam search

$$\vec{y}^* = \arg\max_{\vec{y}} \log P(\vec{y}|\vec{x}) + \lambda \log P_C(\vec{y})$$

E2E Model

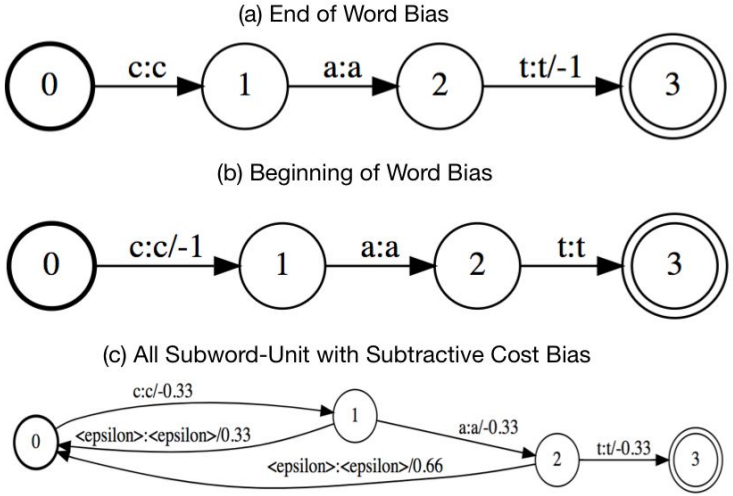
Biasing FST

- Assumptions (for now)
  - Biasing is done at test time only
  - Tune interpolation weight λ per task

Biasing - Where to Apply Scores?

- Best to apply to every unit (E3)

Experiment	Method (Grapheme)	WER - Songs
E0	No Bias (LAS)	20.9
E1	LAS + End of Word Bias	19
E2	LAS + Beginning of Word Bias	16.5
E3	LAS + Every Subword Unit w/ Subtractive Cost Bias	13.4



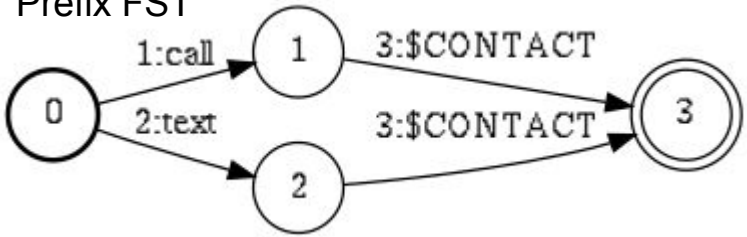
Improving Biasing Further

- Biasing FST should be applied before pruning the beam candidates, not rescoring a pruned beam (E4)
- Biasing at the WPM level is more effective than grapheme (E5)

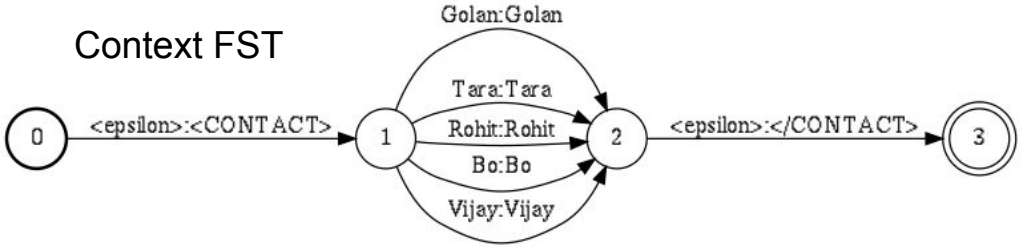
Experiment	Method (Grapheme)	WER - Songs
E0	No Bias (LAS)	20.9
E3	Grapheme Biasing	13.4
E4	Biasing Before Pruning	9.4
E5	4K Word Piece Model LAS Biasing	6.9

Prefixes & Suffixes

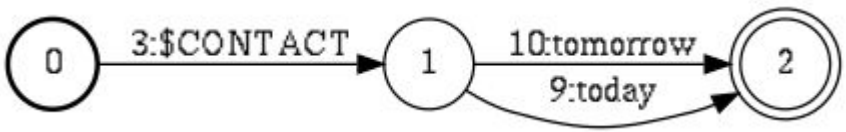
Prefix FST



Context FST

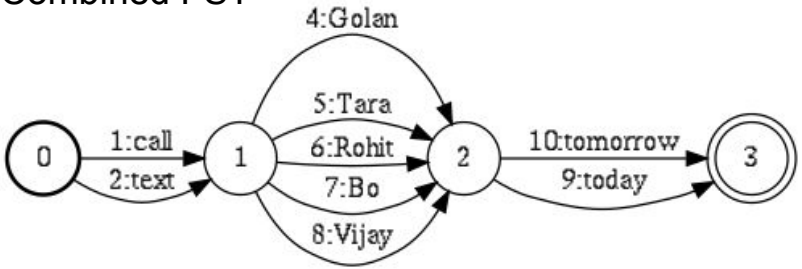


Suffix FST



- Since E2E model cannot predict $CONTACT, we prebuild individual FSTs into one.

Combined FST



Prefixes & Suffixes

- Using this makes a large difference for biasing

Experiment	Method (Grapheme)	WER - Songs
E0	No Bias (LAS)	20.9
E5	4K Word Piece Model LAS Biasing	6.9
E6	+ Prefix and Suffix	5.6

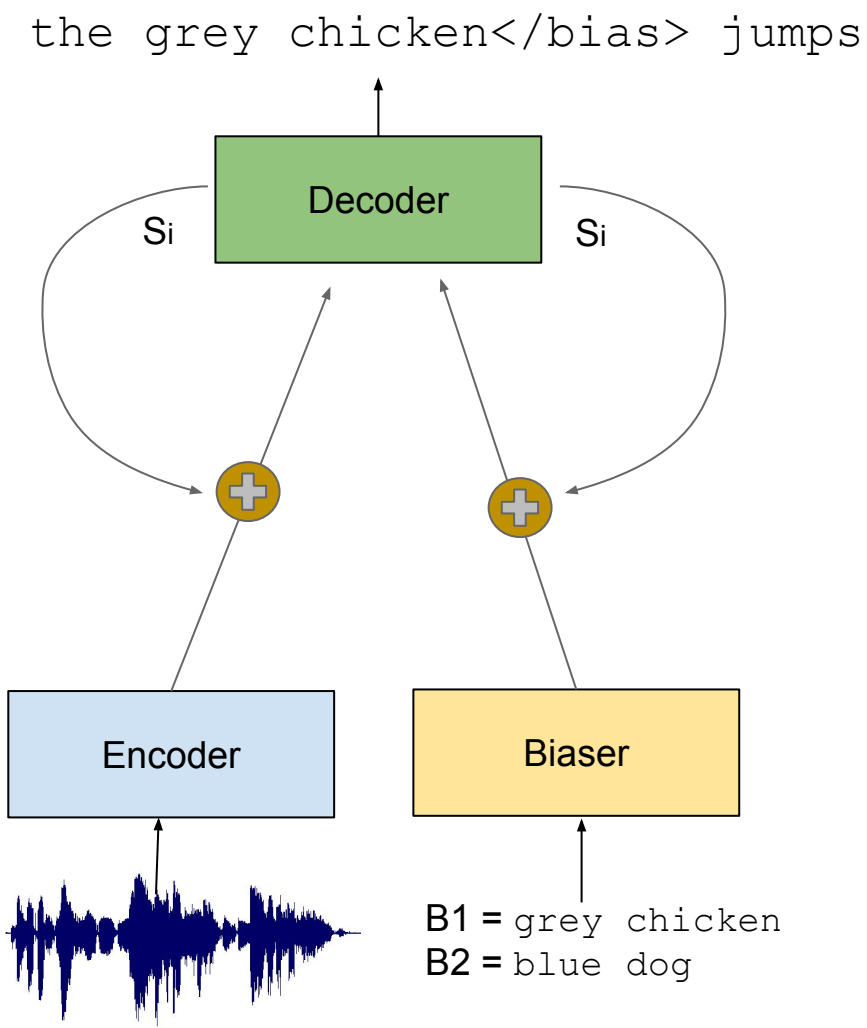
Shallow Fusion Biasing Summary

- Biasing E2E Models similar quality to conventional model

Method	CONTACTS	Songs	THIRD PARTY
Conventional Model No Biasing	36.1	26.5	-
Conventional Model Biasing	10.0	3.8	-
LAS No Biasing	26.9	16.8	10.5
LAS + Shallow Fusion, WPM 4K	7.1	5.6	**3.9**

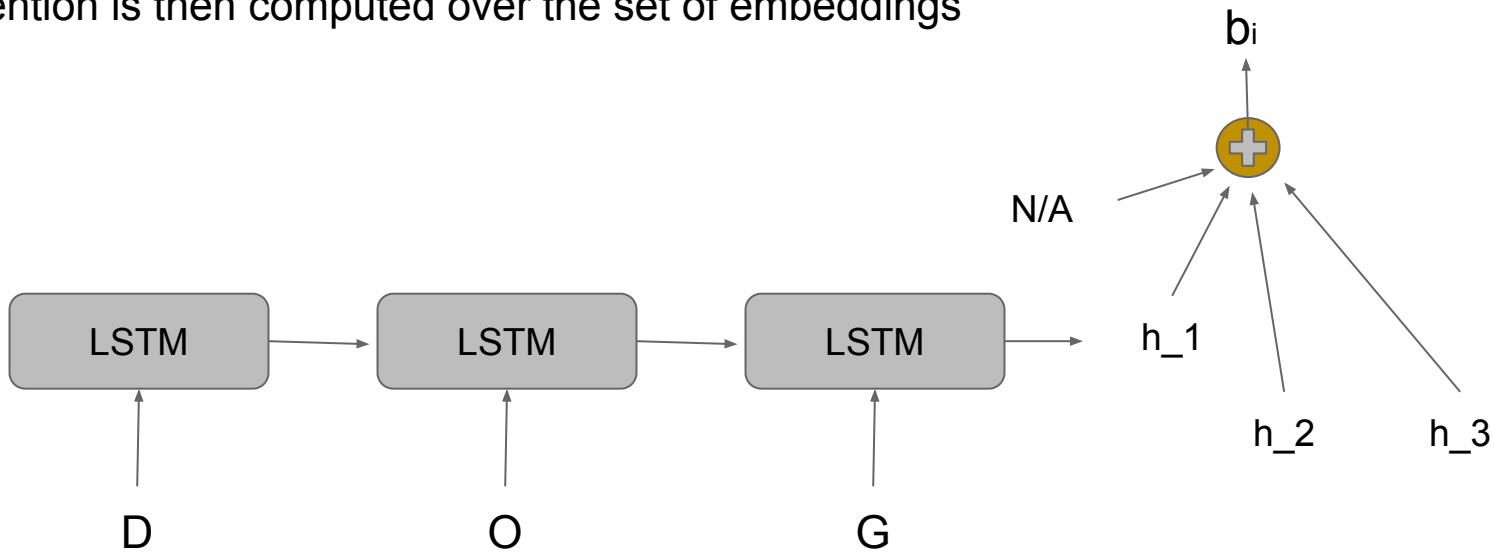
(2) Biased LAS Model (CLAS)

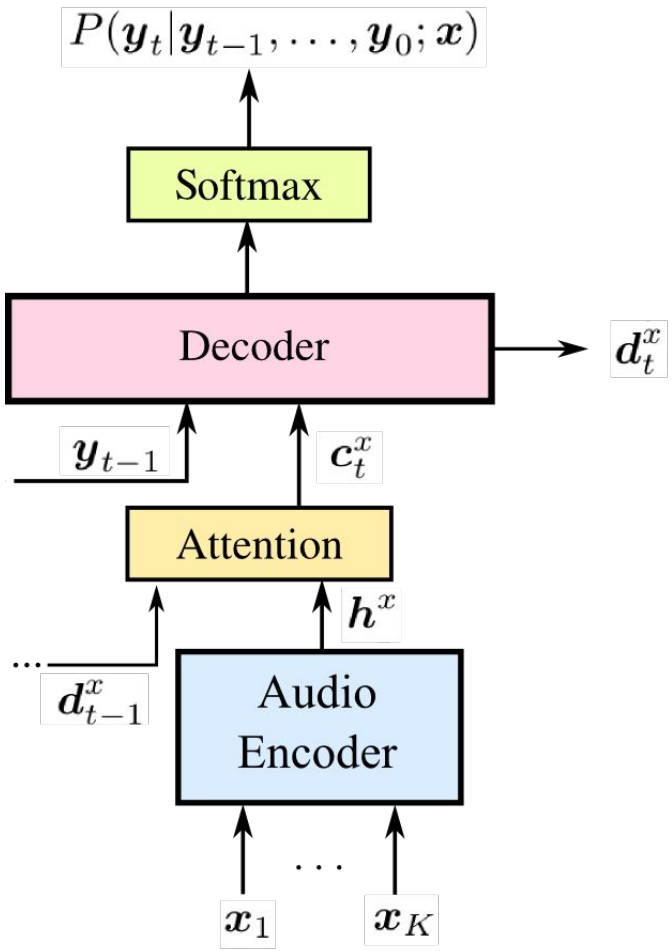
- Fixed-length embedding of bias phrases
- Attention over the embeddings, producing a **bias-dependent per-step** context vector
- Attention also includes a **N/A option** - don't apply bias



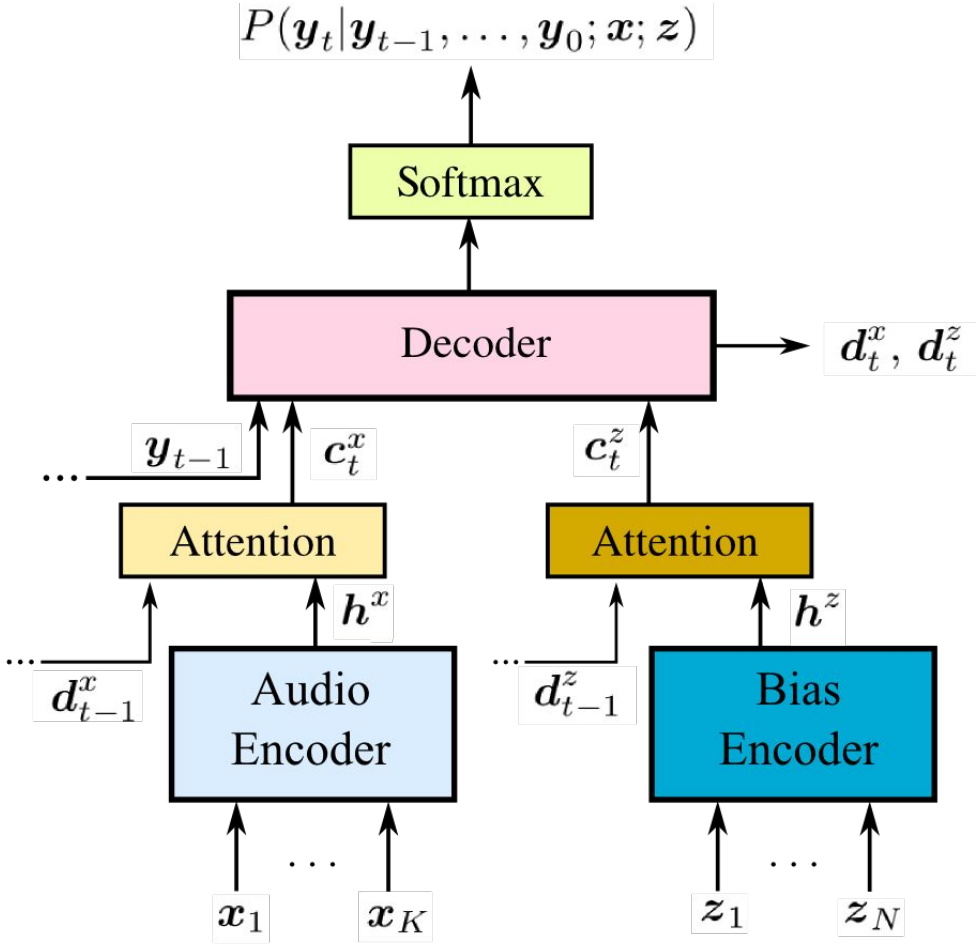
The Biaser

- The Biaser embeds each phrase into a fixed length vector
  - → Last state of an LSTM
- Embedding happens once per bias phrase (possibly offline)
  - Cheap computation
- Attention is then computed over the set of embeddings





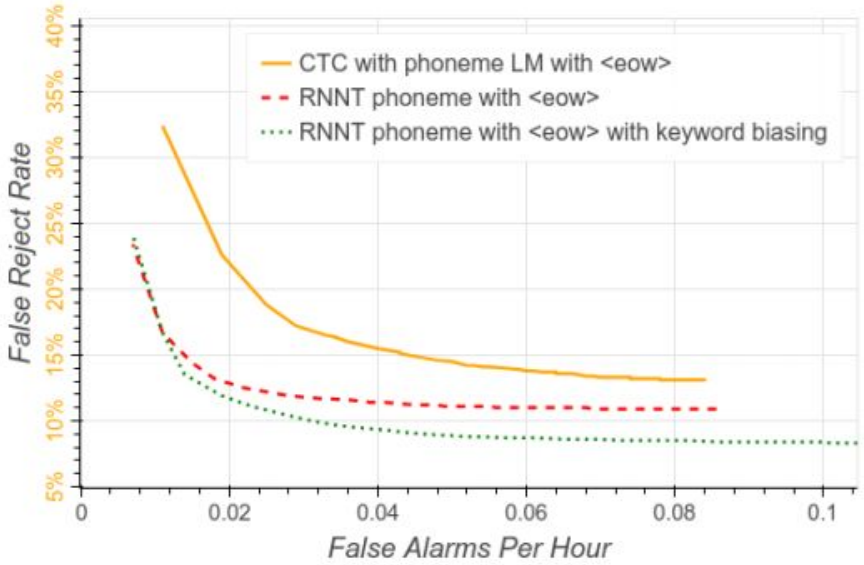
(a.) LAS Model

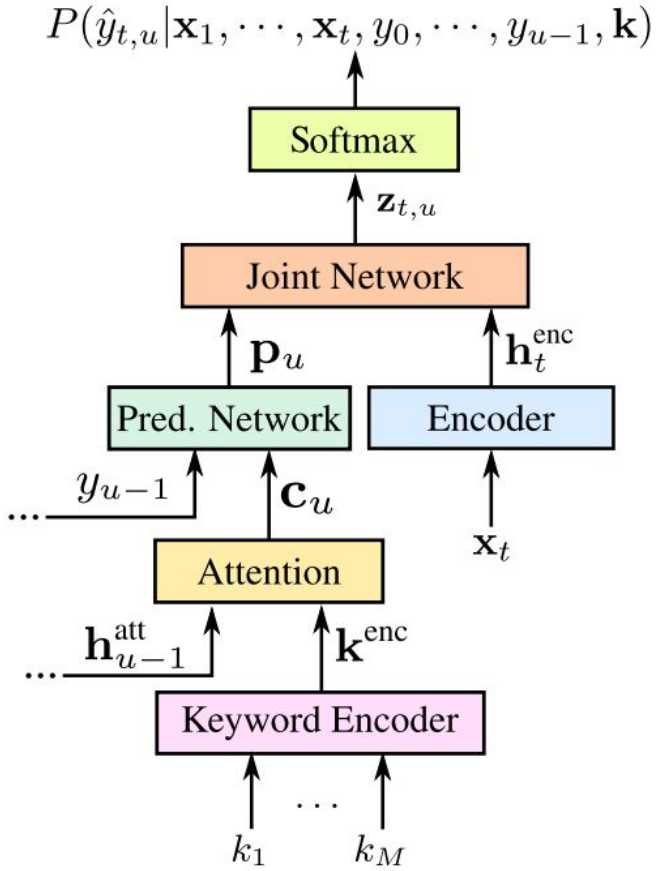


(b.) BLAS Model

Prior work: Keyword spotting with RNNT

- “Streaming Small-Footprint Keyword Spotting using Sequence-to-Sequence Models” [He et al., 2017]



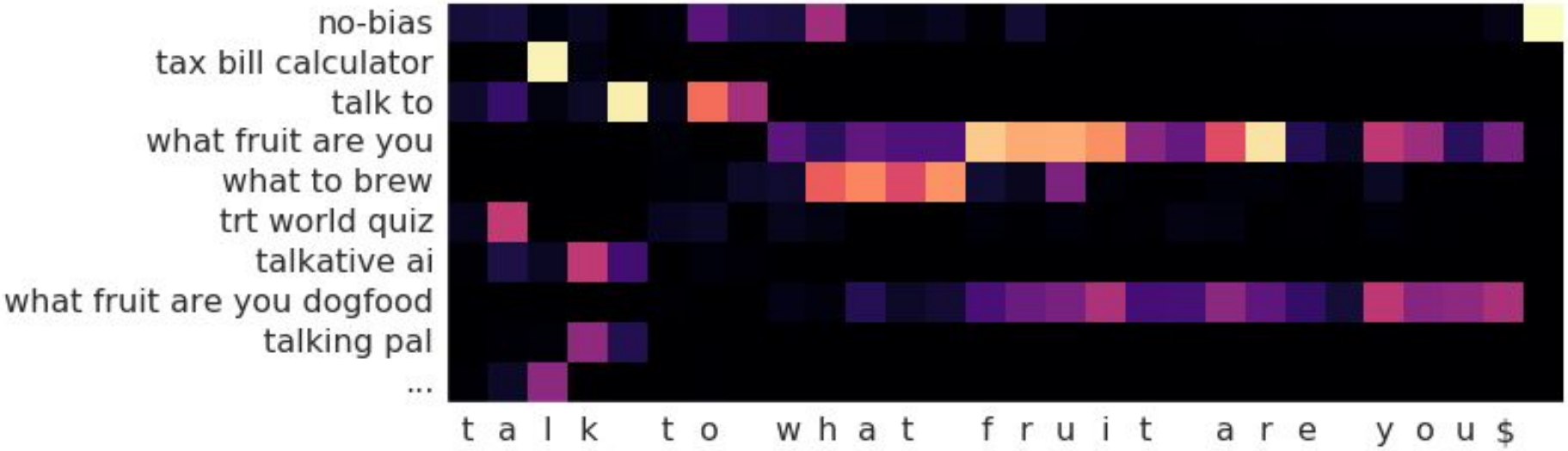


(c.) RNN-Transducer Biased with Attention Over Keyword.

CLAS training

- Example ref: `The grey chicken jumps over the lazy dog`
- Sample uniformly a bias phrase ***b***, e.g. `grey chicken`
- With **drop-probability** p (e.g. 0.5) drop the selected B and replace it with another bias phrase from the same batch
- Augment with additional N-1 more bias phrases from other references in the batch (distractors)
- Present the model the set of N (shuffled) bias phrases:
  - `quick turtle`
  - `grey chicken`
  - `brave monkey`
- If ***b*** was not dropped, insert a `</bias>` token to reference:
  - `The grey chicken</bias> jumps over the lazy dog`

Biasing Example



Key aspects of CLAS

- Biasing is viewed as a keyword detection task which relates to both audio and LM (cf. beam search biasing)
- CLAS embeds “long” var-length bias sequences into fixed-length vectors
- CLAS computes attention over a set of phrases
- The model can take any list of bias phrases in inference time (including OOVs)
  - In training the bias phrases list is randomized for each batch
  - The number and content of bias phrases can be changed from training to inference

Biasing Summary

- CLAS model performs similar to biasing of conventional model

Method	CONTACTS	Songs	THIRD PARTY
Conventional Model No Biasing	36.1	26.5	-
Conventional Model Biasing	10.0	3.8	-
LAS No Biasing	26.9	16.8	10.5
CLAS + Shallow Fusion, Grapheme	7.5	5.7	5.6