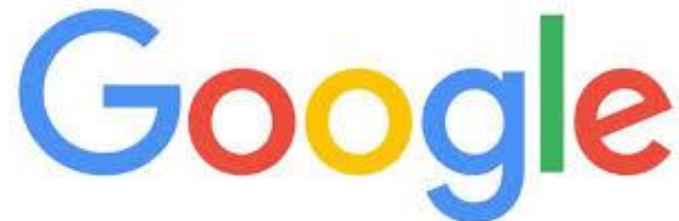




Uniwersytet
Wrocławski



Deep neural networks for speech and natural language processing

Jan Chorowski

University of Wrocław

Visiting Researcher at Google Brain

Collaborators

Much of my research was done with friends from
Universite de Montreal, Uniwersytet Wrocławski,
and Google

- Dzmitry Bahdanau
- Dmitriy Serdyuk
- Philémon Brakel
- Kyung Hyun Cho
- Yoshua Bengio
- Adrian Łańcucki
- Michał Zopotoczny
- Paweł Rychlikowski
- Szymon Malik
- William Chan
- Yu Zhang
- Navdeep Jaitly
- Ron Weiss
- Samy Bengio

Outline

- End-to-end speech recognition systems
- Challenges and solutions
- Speech Translation
- Attention mechanism and NLP

Classical ASR and NLP Pipelines

Many separately trained parts:

- Speech recognition:
 - Feature extraction, spelling lexicon, acoustic model, alignment model, language model.
- Translation:
 - Language model, alignment model, translation table.
- Speech translation:
 - All the parts of ASR and translation systems!
- Parsing, sentence understanding:
 - Morphosyntactic dictionary, tagger, parser.

End-to-end systems

What is end-to-end:

- *“training all the modules to optimize a global performance criterion”*
(“Gradient-based learning applied to document recognition”, LeCun et al., 98)
system for recognizing checks in which segmentation and character recognition are trained jointly with word constraints taken into account (the approach would now be called Conditional Random Fields)

Not end-to-end: hand-crafted feature engineering, manual integration of separately trained modules.

Why end-to-end: better performance, better portability, system is easier to manage, model doesn't have to commit early to a decision.

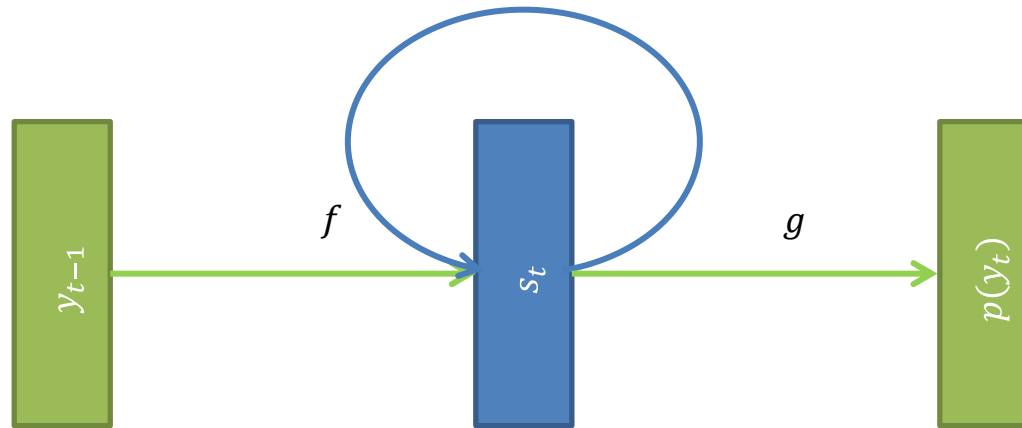
End-to-end systems are here

- Convolutional networks for object recognition (Krizhevsky et al., 12)
- Neural Machine Translation: take raw words as the input, all components trained together (Sutskever et al., 14, Bahdanau et al., 15)
- Neural Caption Generation: produce image descriptions from raw images (Vinyals et al. 2014, Xu et al. 2015)
- Proof of concept Speech Translation (Weiss et al. 2017)

Design of an end-to-end System

- Reduce the problem to supervised learning.
- Directly model $p(Y|X)$
where Y : desired output (e.g. characters)
 X : inputs (e.g. speech frames)
- Build generic systems:
 - Inputs X can be acoustic frame, character, word **sequences**
 - Outputs Y can be character, word, labeled graph edge **sequences**
 - The X and Y sequences need not be aligned.

RNNs Learn $p(Y)$



Decompose

$$p(Y) = \prod p(y_t | y_{t-1}, y_{t-2}, \dots, y_1)$$

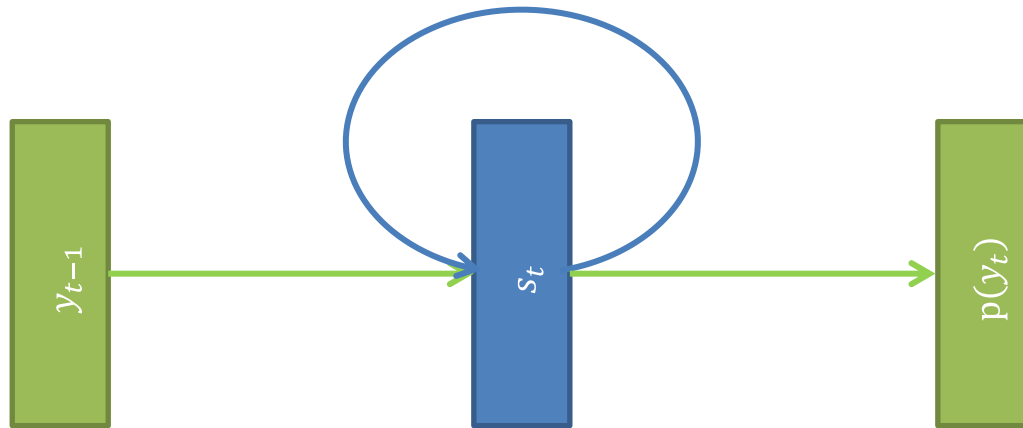
Model the probabilities using a recurrent relation

$$p(y_t | y_{t-1}, y_{t-2}, \dots, y_1) = g(s_t)$$

$$s_t = f(s_{t-1}, y_{t-1})$$

$g()$, $f()$ are implemented using neural networks, i.e. they are flexibly parameterized, smooth functions.

How to condition an RNN?

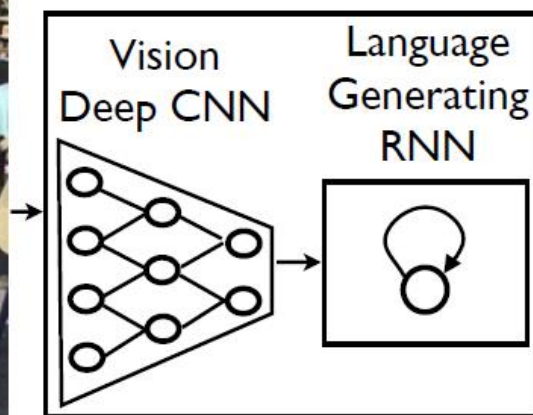


RNN gives us $p(Y)$ but we want $p(Y|X)$

- Idea #1: conditioned through the first hidden state
- Idea #2: condition separately on every step

Idea #1

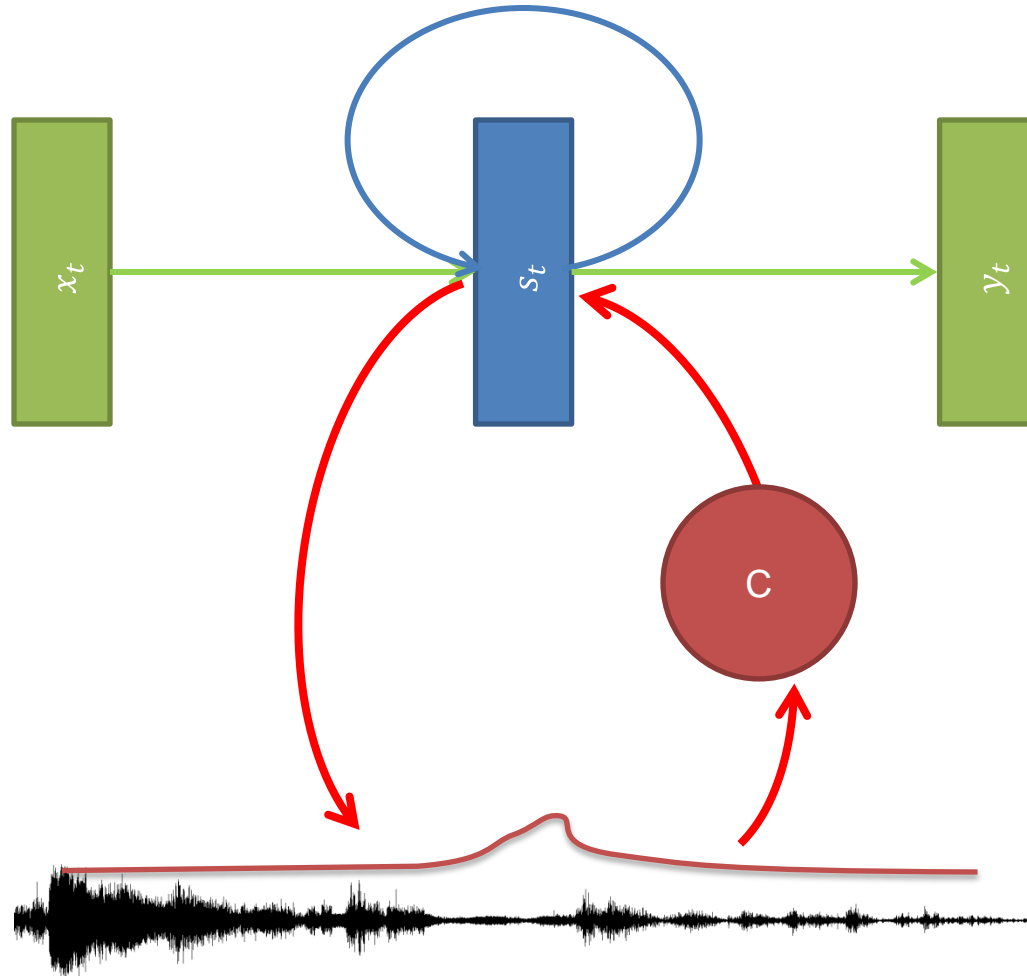
condition through the 1st hidden state



A group of people shopping at an outdoor market.

There are many vegetables at the fruit stand.

Idea #2: Attention



1. Choose relevant frames

$$e_f = \text{score}(x_f, s_{t-1})$$

$$\alpha_f = \text{SoftMax}(e)_f$$

2. Summarize into context

$$c = \sum_f \alpha_f x_f$$

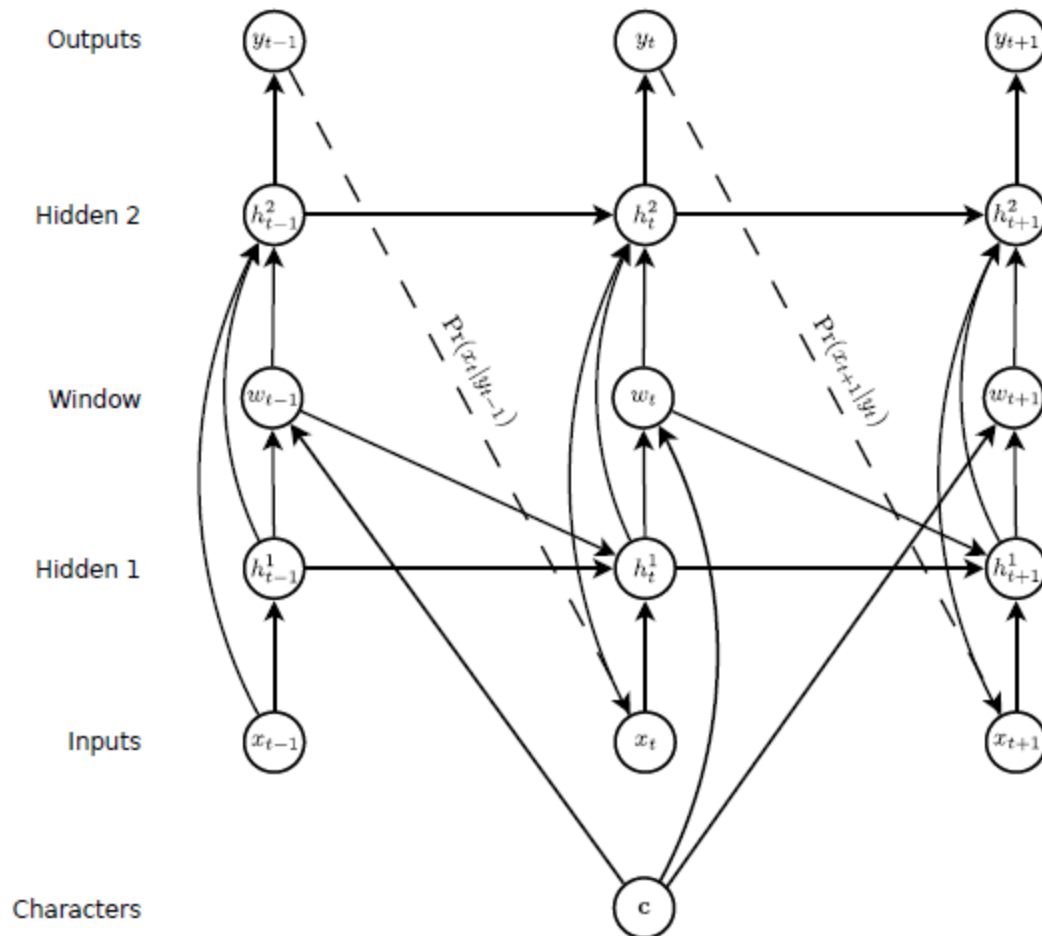
3. Compute next state

$$s_t = f(s_{t-1}, y_{t-1}, c)$$

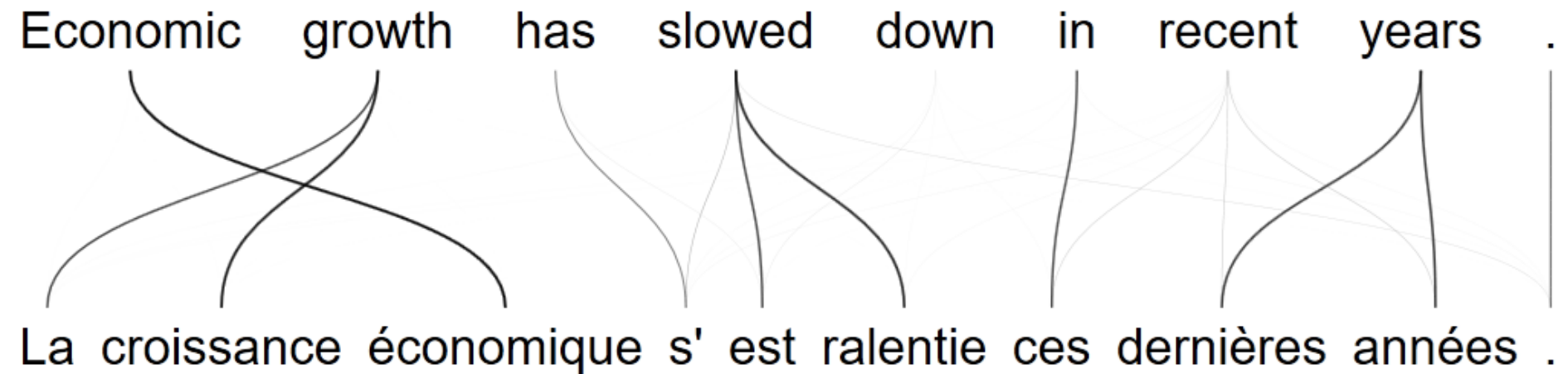
Attention mechanism in RNNs

- This is a network to generate handwriting
- At each step the network looks at a *context c*
- c is a summarization of a small fragment of the input sequence

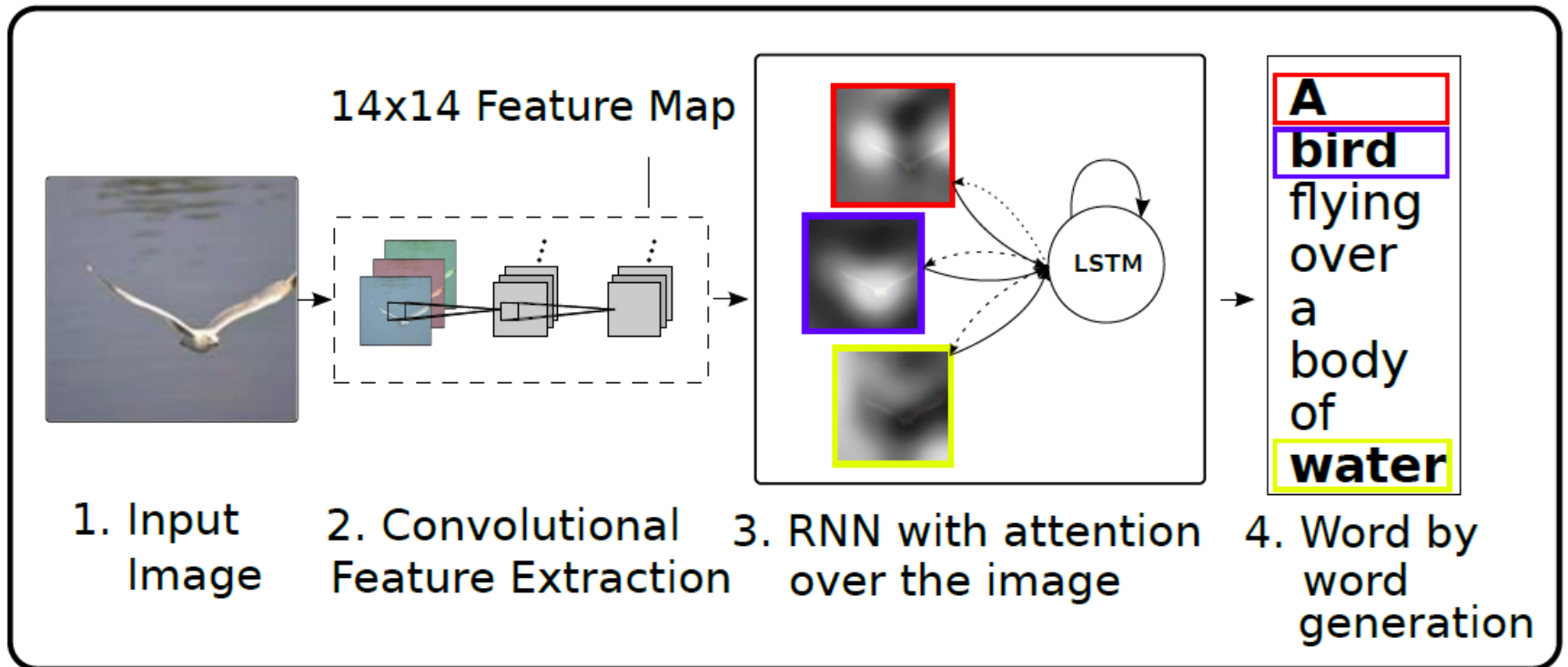
from his travels it might have been
from his travels it might have been
from his travels it might have been



Attention mechanism in translation



Attention mechanism for captioning



Typical Attention ASR at a Glance

Network defines

$$p_{\Theta}(Y|X)$$

where

$$Y = y_1 y_2 \dots y_T$$

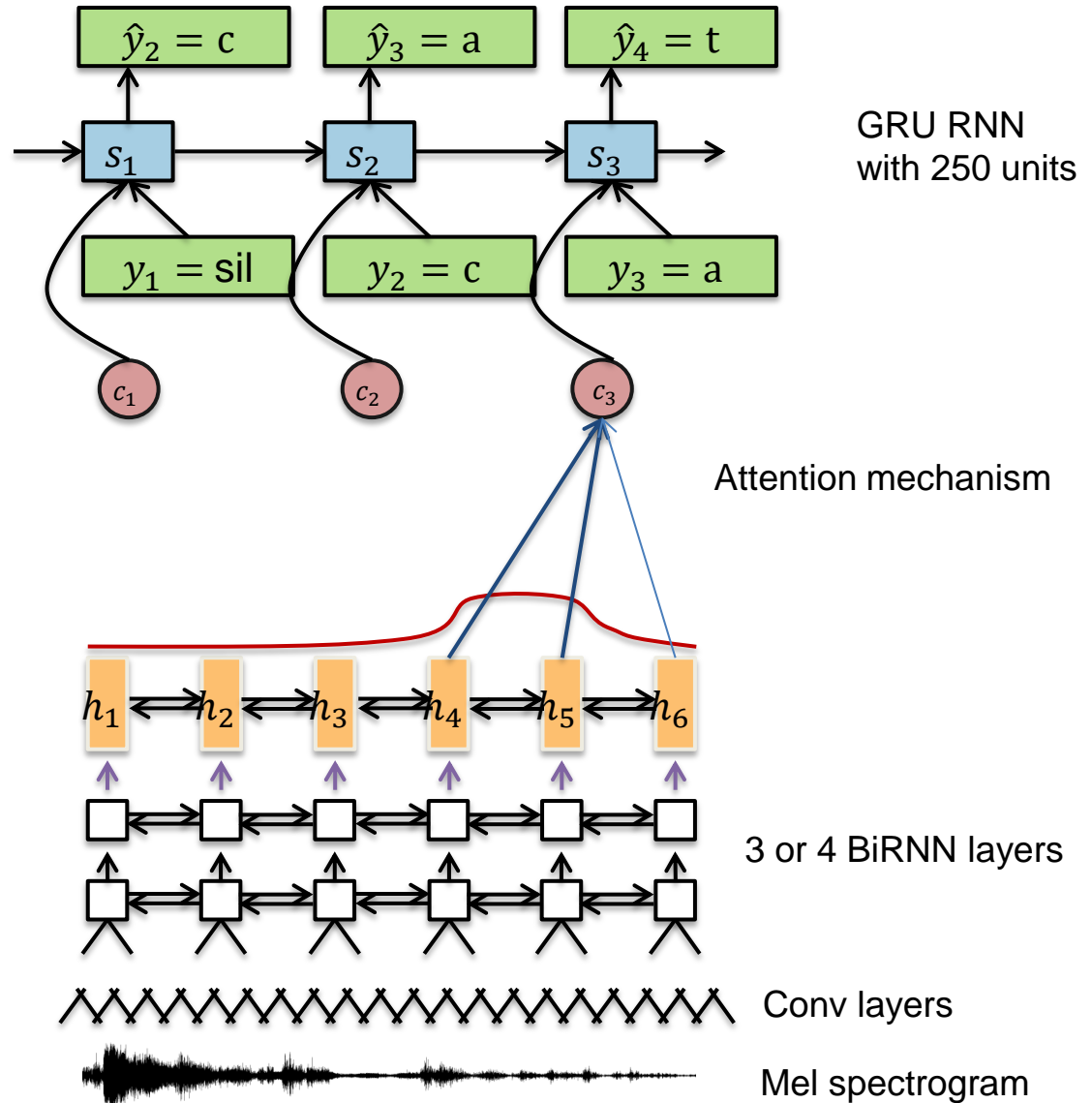
are characters

$$X = x_1 x_2 \dots x_F$$

are speech
frames

Θ are

parameters



Training

We train the network to maximize the log-likelihood of the correct output

$$\frac{1}{N} \sum_{i=1}^N \log p_{\Theta}(Y_i | X_i)$$

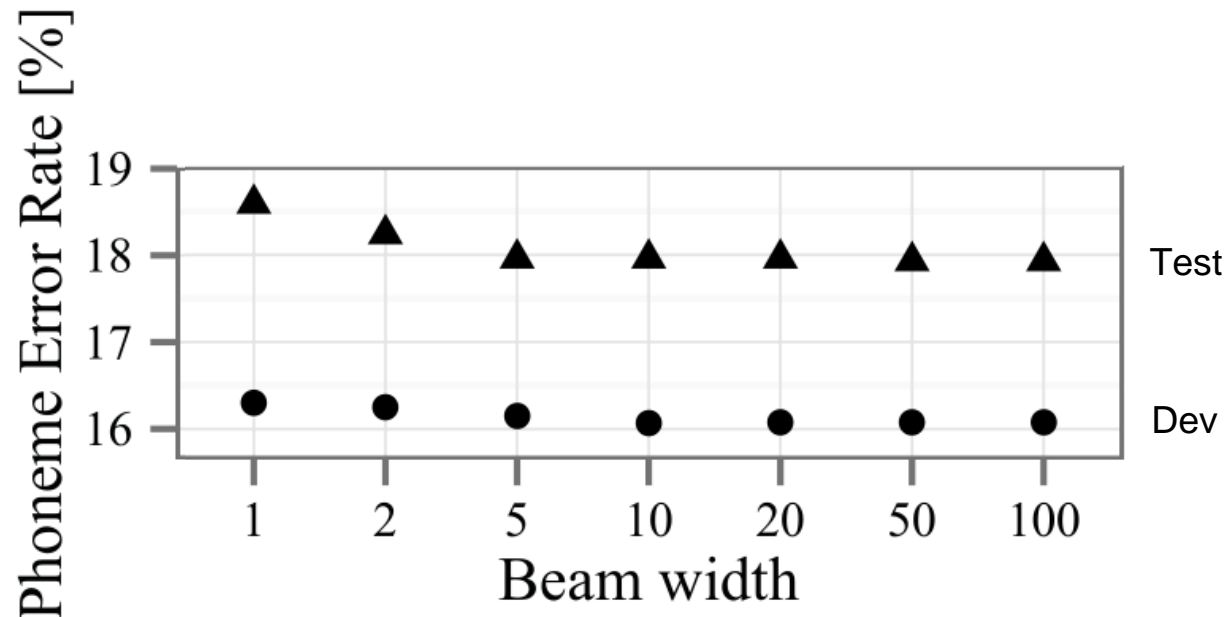
p_{Θ} is differentiable with respect to Θ thus we can use gradient based methods.

Decoding

Use beam search to find

$$\hat{Y} = \arg \max_Y p_{\Theta}(Y|X)$$

Narrow beams are needed:



Tricks of the Trade: Regularization

RNNs don't like weight decay too much

- Small weights, especially small recurrent weights mean signal decay

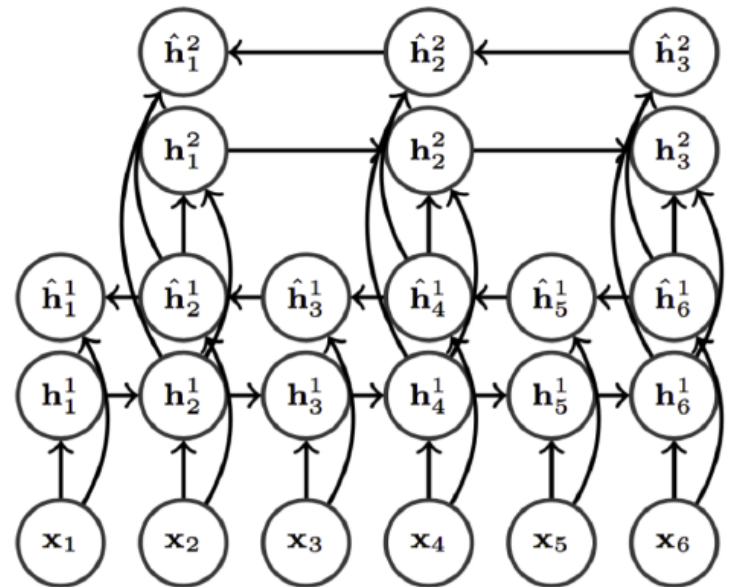
Better add noise to weights

- Weights can be large
- But low-precision
- Promotes wide minima (small change to weights doesn't matter!)

Tricks of the Trade: Subsampling

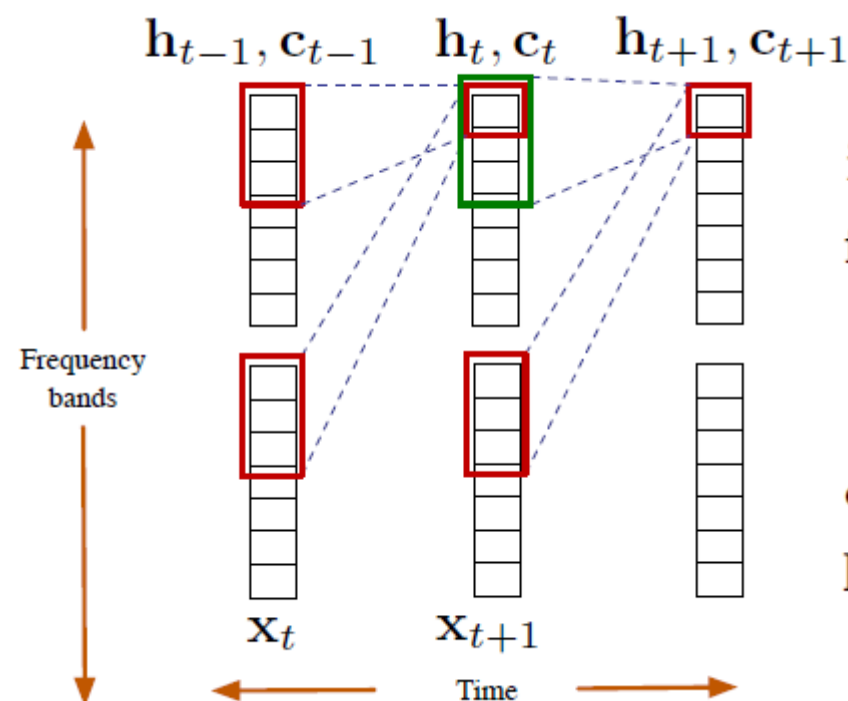
BiRNN encoders with subsampling

- BiRNN = forward RNN + backward RNN (both without outputs)
- Deep BiRNN = states of the layer K are the inputs of the layer $K + 1$



Tricks of the trade: ConvLSTM

Replace matrix multiplications with convolutions!
Better preserves time-frequency structure.

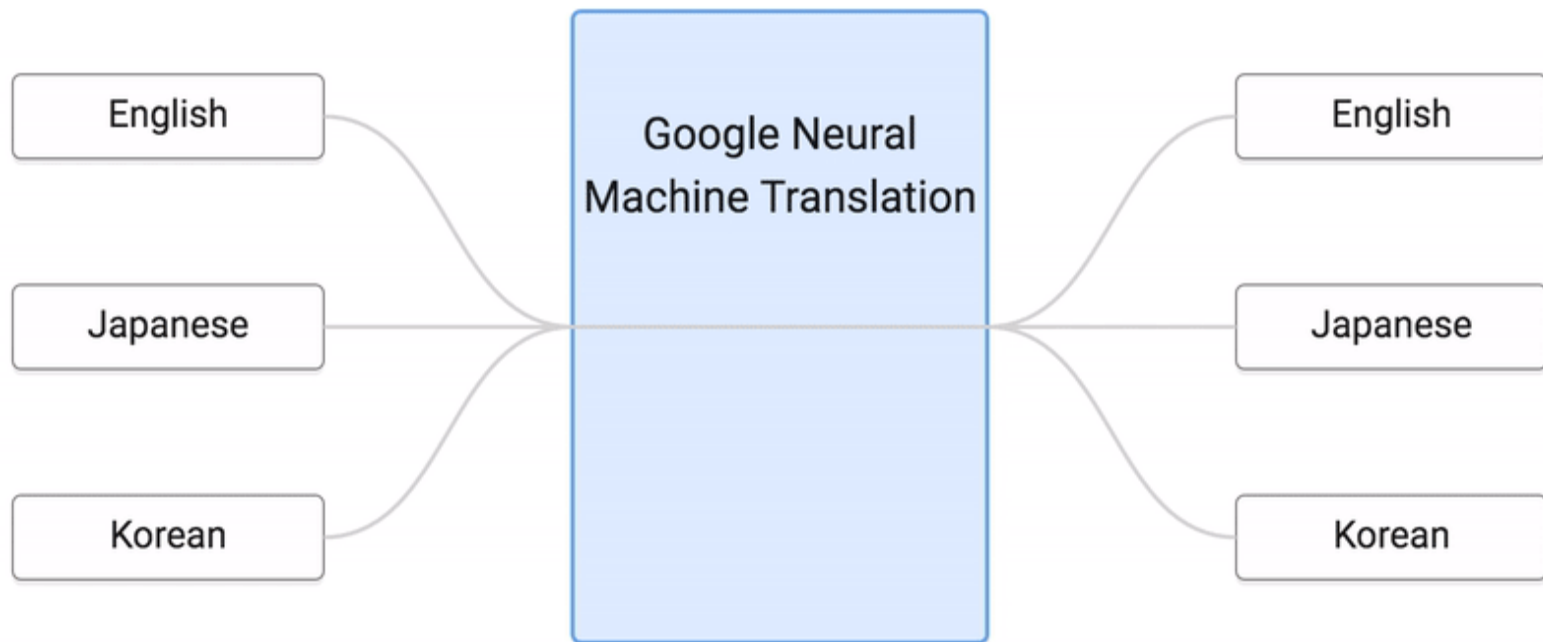


$$\begin{aligned}i_t &= \sigma(W_{xi} * x_t + W_{hi} * h_{t-1} + b_i) \\f_t &= \sigma(W_{xf} * x_t + W_{hf} * h_{t-1} + b_f) \\c_t &= f_t \odot c_{t-1} + \\&\quad + i_t \odot \tanh(W_{xc} * x_t + W_{hc} * h_{t-1} + b_c) \\o_t &= \sigma(W_{xo} * x_t + W_{ho} * h_{t-1} + b_o) \\h_t &= o_t \odot \tanh(c_t)\end{aligned}$$

Tricks of the trade: Multitask

- Multilingual translation: zero-shot translation

Training

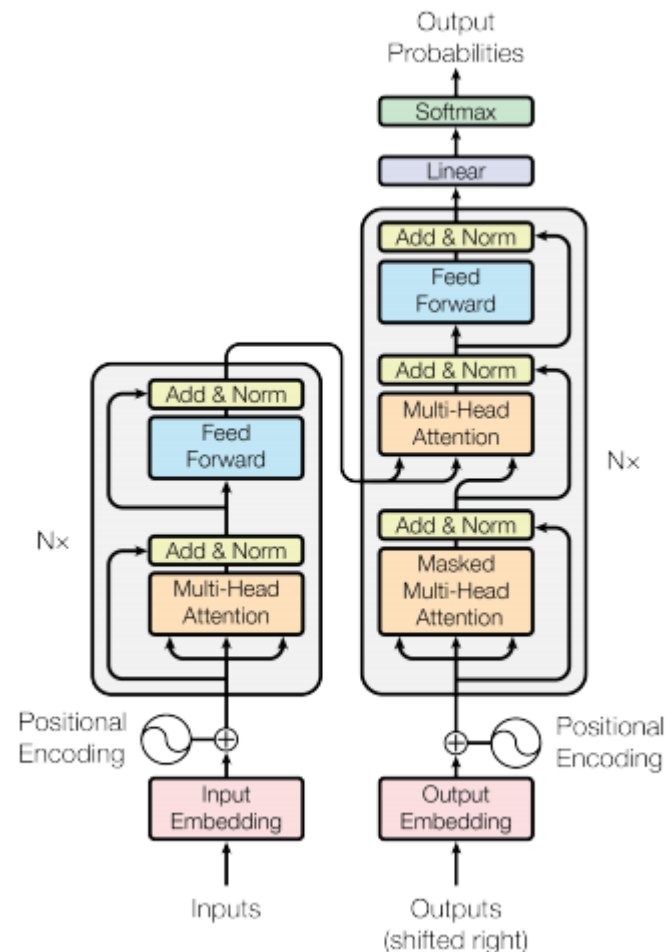
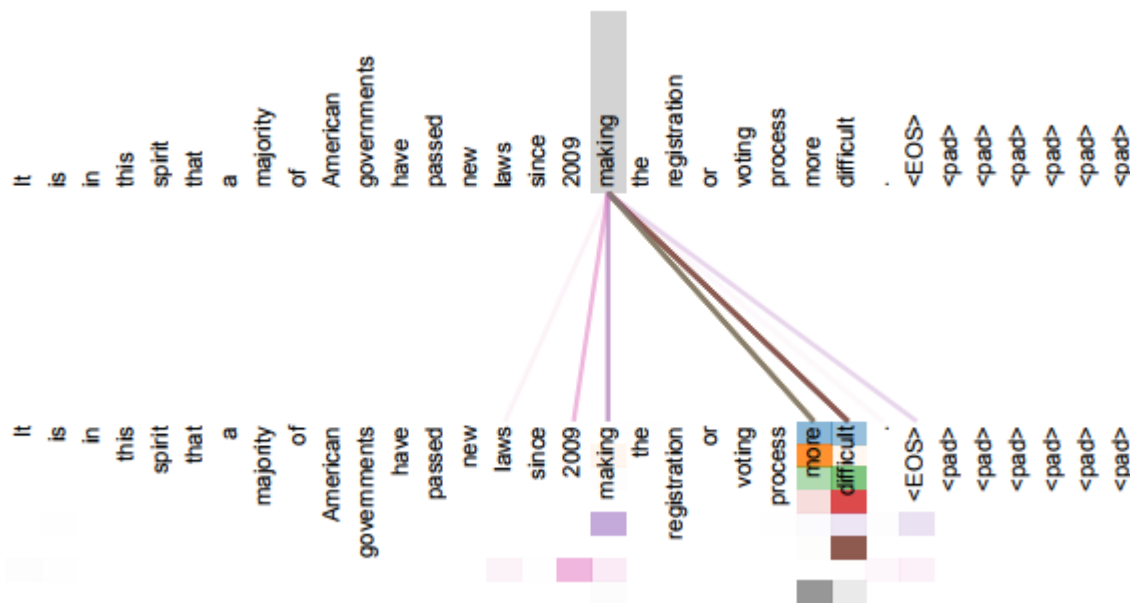


New developments: Attention is All You Need

RNN: compress history into the state vector

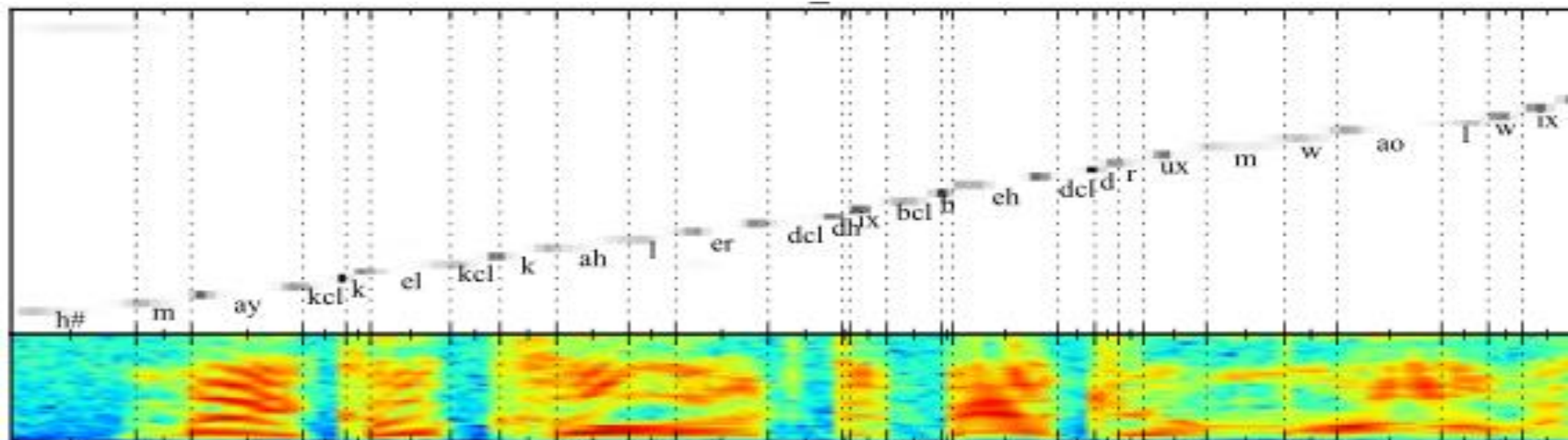
UniRNN: attention over history!

BiRNN: attention over whole sequence



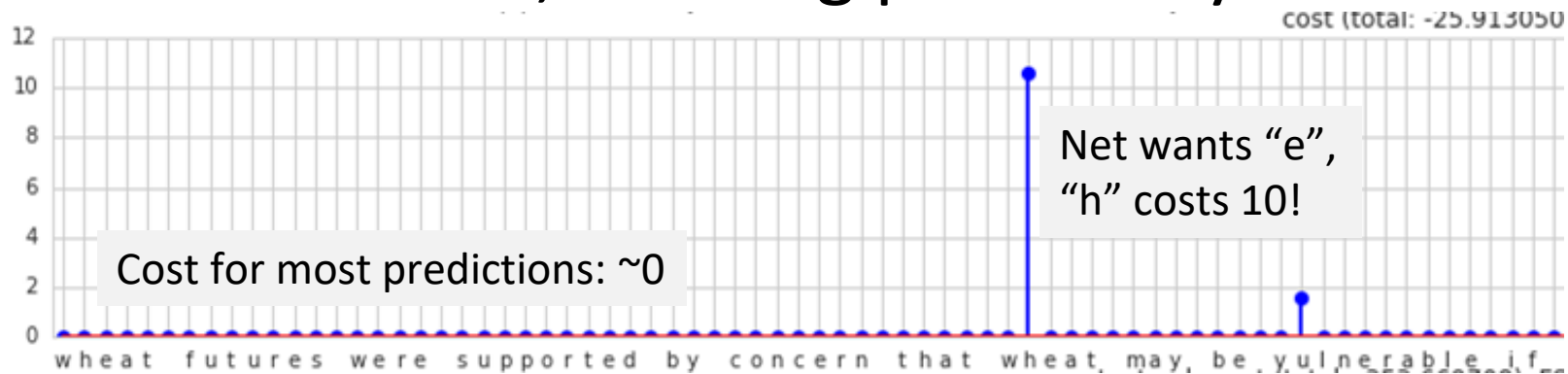
Challenges

- Overconfidence.
- Long sequences and repetitions.
- Language model integration and coverage.

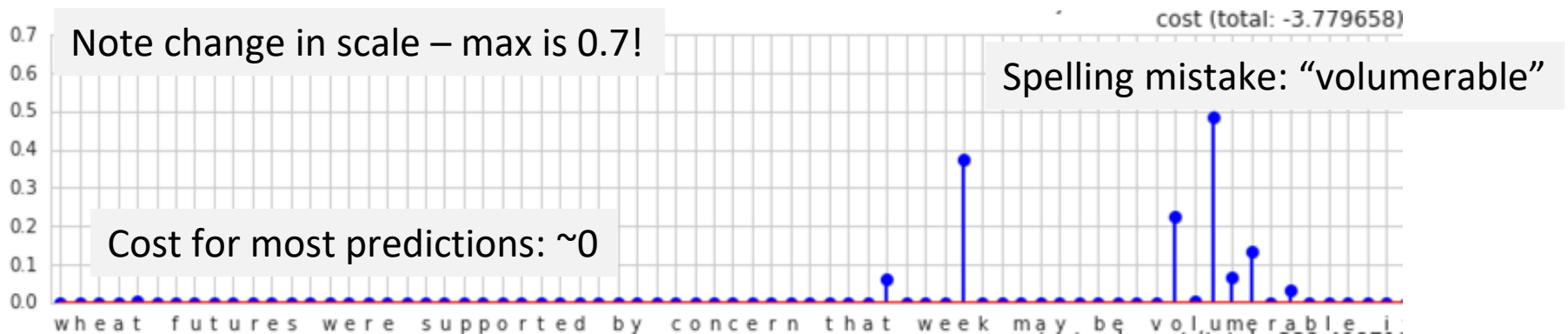


Overconfidence

Ground truth, total log probability -25



Beam search result: total log probability -3.7



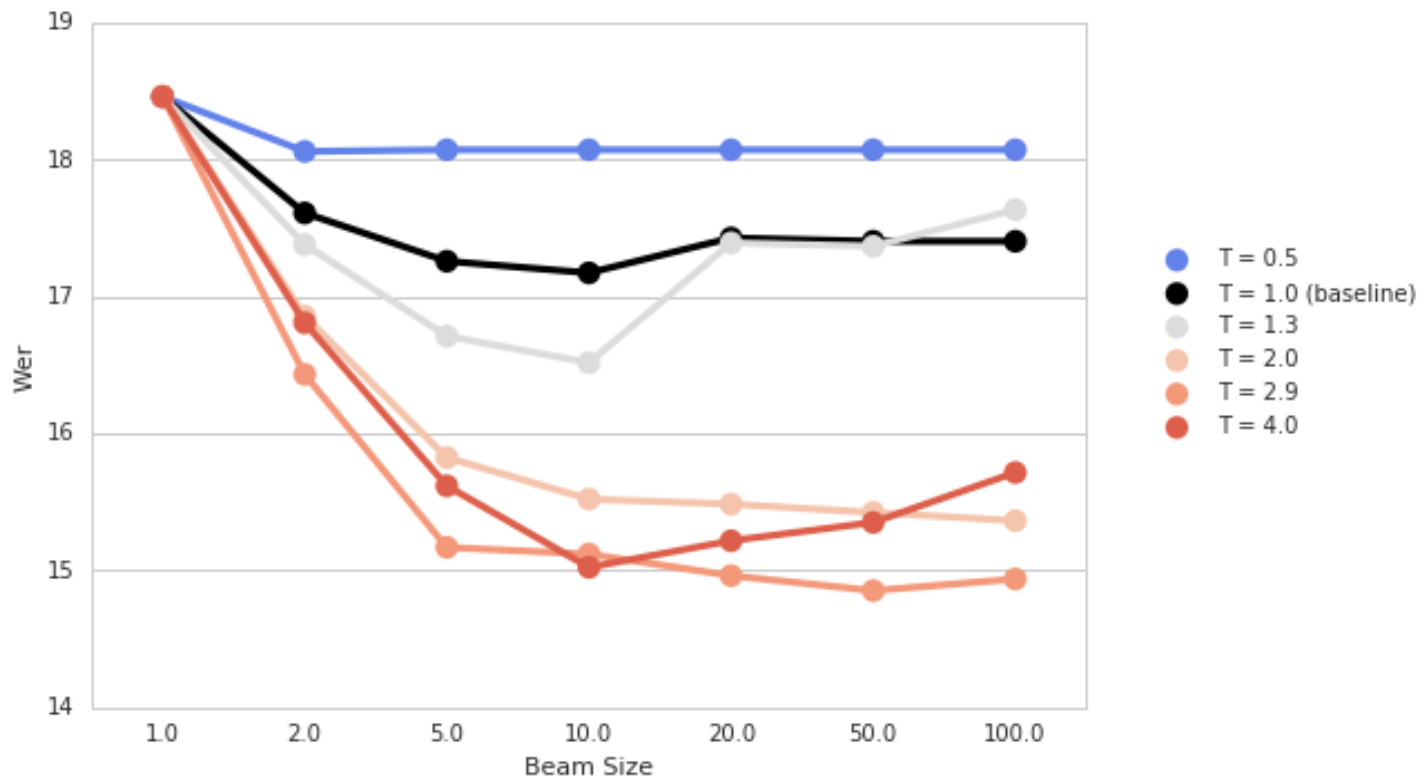
Key Observations

- Accurate next-step predictions:
99.9% train/96% test
- Overconfidence:
 $p(\text{first guess}) \gg p(\text{second guess})$
- A “second guess” of the net costs as much as several “first guess” predictions
 - Beam search ineffective at large beams
 - Very hard to balance decoding costs (e.g. LM)

A Simple Experiment

- After training, tweak SoftMax temperature

$$\text{SoftMax}(Y) = \frac{\exp(Y_i/T)}{\sum_j \exp(Y_j/T)}$$



Training With 1-hot Labels

- The cross-entropy cost for one utterance


$$-\sum_{i=1}^N \sum_c [Y_i = c] \log p_{\Theta}(Y_i | Y_{<i}, X_i)$$

- When model is 99% accurate...
- The only way to reduce cost is to make $p_{\Theta}(Y_i | Y_{<i}, X_i)$ a Dirac delta...

Training With Label Smoothing

- Introduced in Inception V2 (arXiv:1512.00567)

- Change the cost to:

$$-\frac{1}{N} \sum_{i=1}^N \sum_{c=1}^C T(Y_i, c) \log p_{\Theta}(Y_i | X_i)$$


$[Y_i = c]$

- $T(Y_i, c)$ is a smoothing distribution, e.g.

$$T(Y_i, c) = \begin{cases} \beta, & \text{when } Y_i = c \\ \frac{1 - \beta}{C - 1}, & \text{otherwise} \end{cases}$$

- Even better: smooth the $1 - \beta$ according to class marginal probabilities (unigrams)

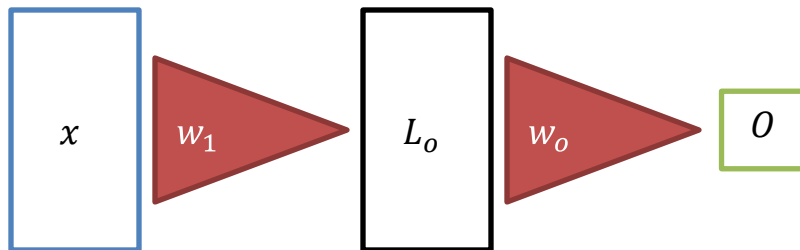
Effects of Label Smoothing

- Reduces overconfidence and regularizes
- Also prevents gradient vanishing:
 - Without smoothing SoftMax derivative is $p_{\Theta}(Y_i|X_i) - [Y_i = c]$
 - This vanishes when $p_{\Theta}(Y_i|X_i) \approx 1$
 - Effectively the model stops training on correctly classified characters

Label Smoothing vs Other Regularizers

At a high level, all regularizers want to forbid large changes of output for small changes of input.

- E.g. weight decay

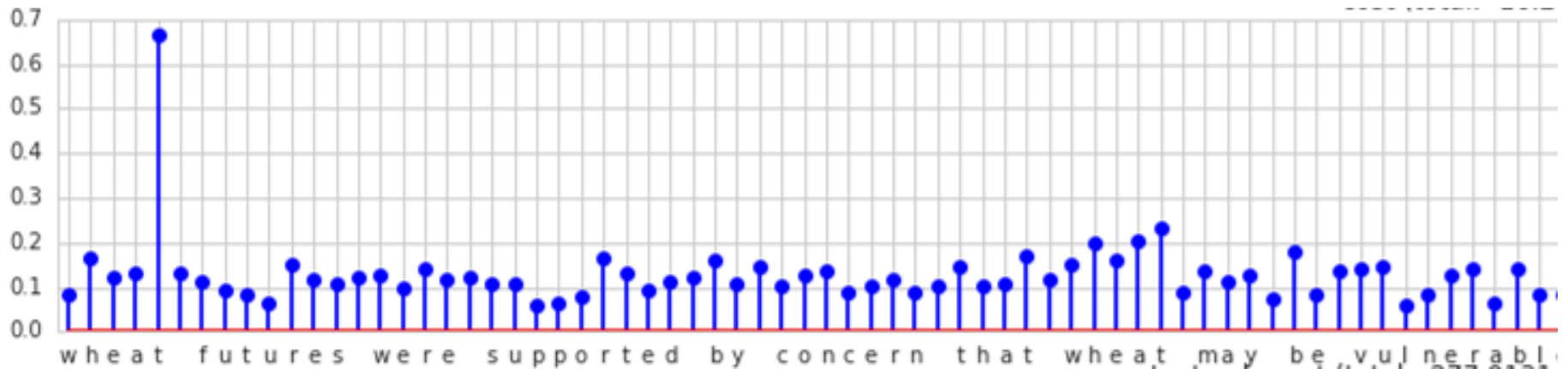


Magnitude of w_o controls the output sensitivity $\frac{\partial o}{\partial L_o} = w_o^T$

- Label smoothing may be easier to use:
 - Easy to say how smooth the output should be
 - Hard to say how large the weights should be

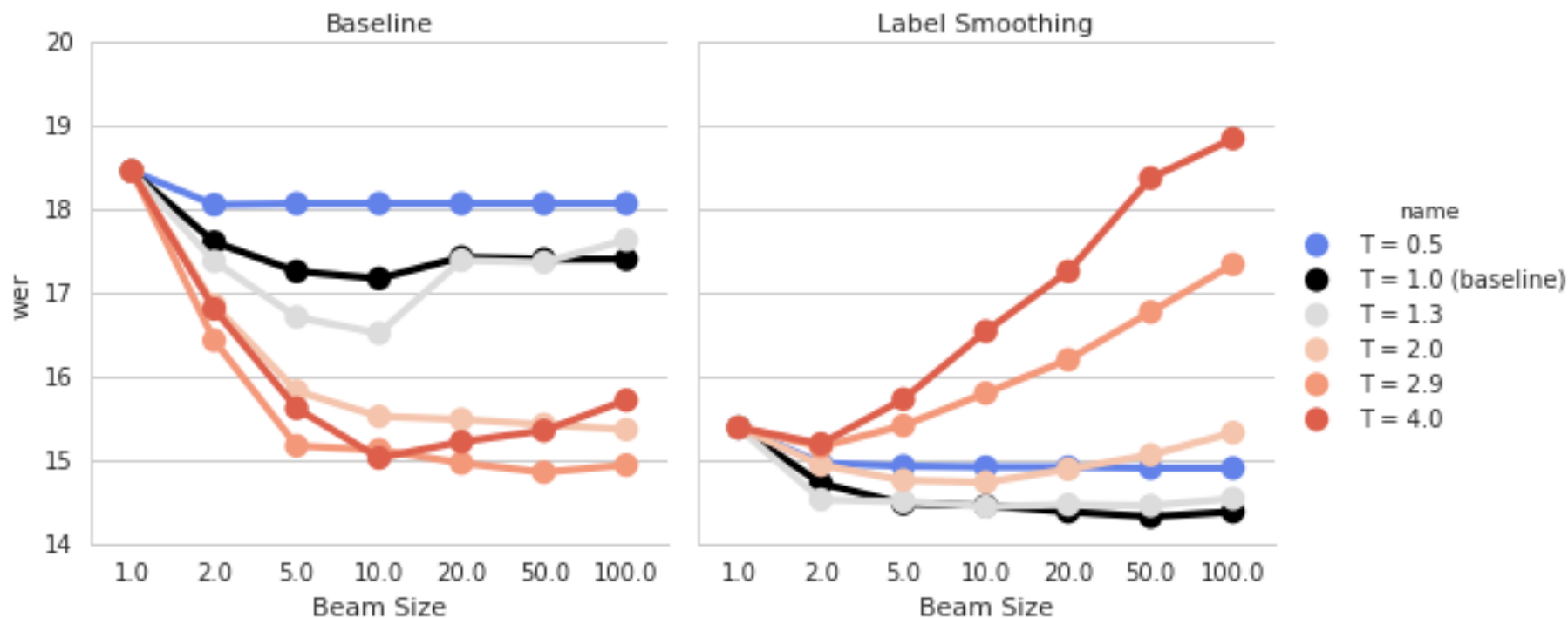
Effects of Label smoothing

- Regularization (next character accuracy increase 96% -> 97%)
- Increase of neg log-probability of best predictions -> other costs easier to balance



SoftMax Temperature and Label Smoothing

- Temperature tweaking no longer needed:



Trouble With Long Sequences

A simple experiment:

1. Train a network as usual.
2. Concatenate test utterances a few times.
3. Decode as usual.

Performance drops dramatically.

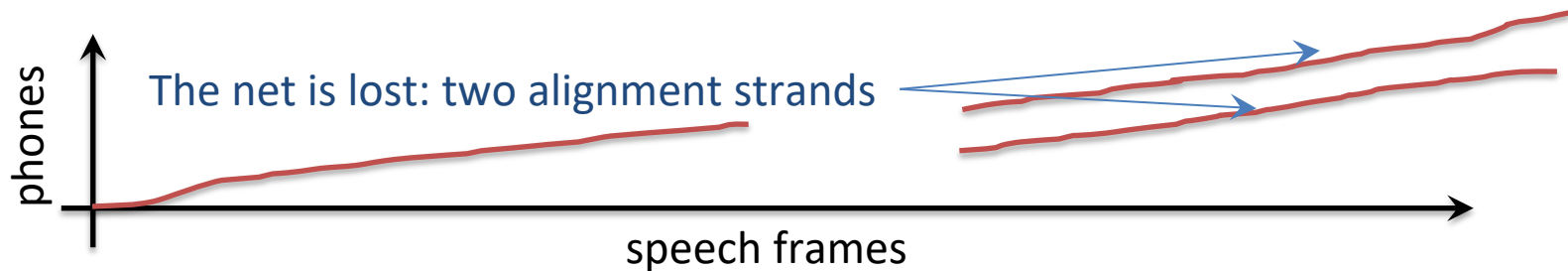
On long utterances decoding completely fails.

Investigation of Long Inputs

The setup:

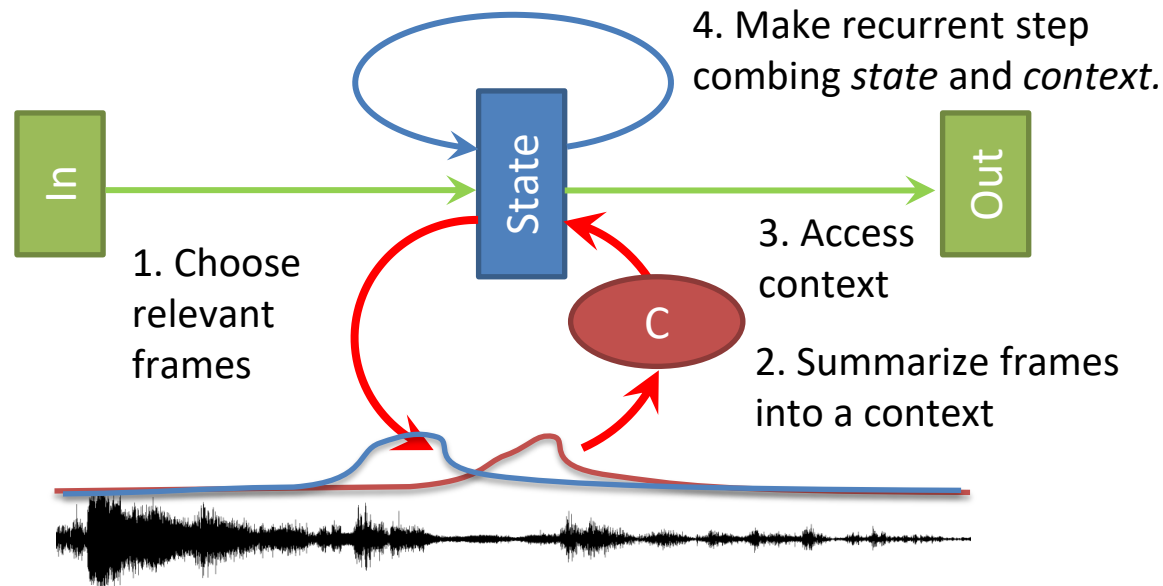
- concatenate utterances
- do force alignment (feed the correct inputs)

Typical result



Our hypothesis: the net learns an implicit location encoder. It is not robust to long utterances.

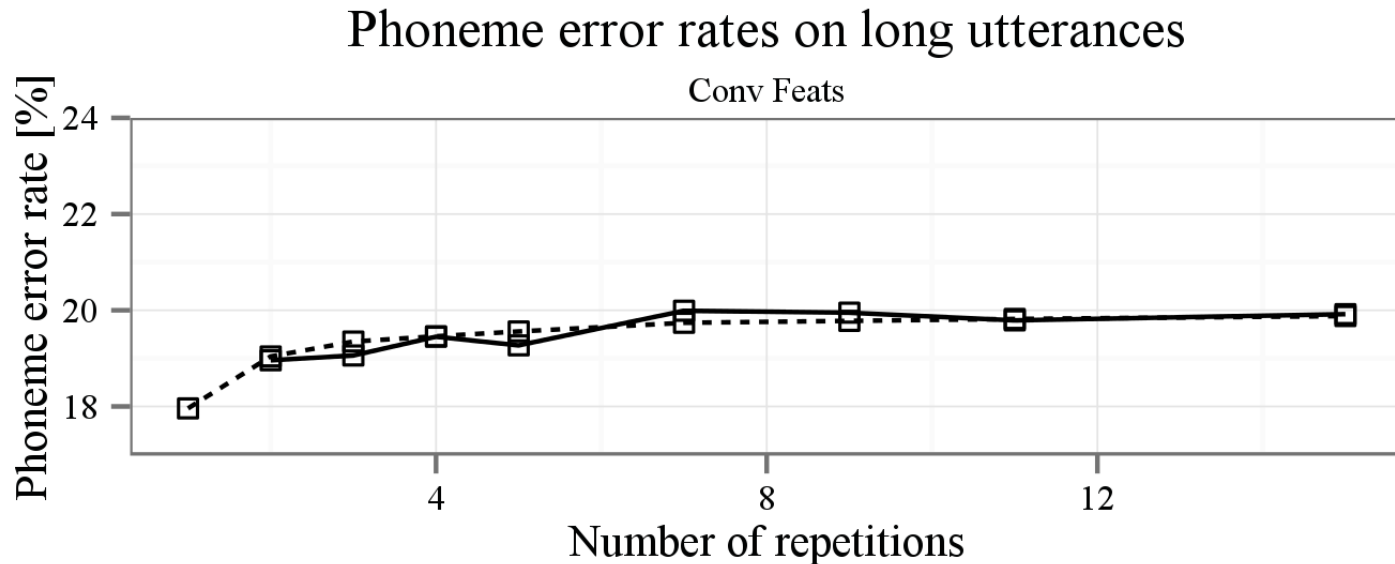
Location-aware Attention



- We want to separate repetitions of the same sound
- Use the selection from the last step to make the new selection
- This enables the model to learn concepts like “later than last” or “close to last”.

Location-aware attention helps

- Decoding error rate increases from 18% to 20%



- One more “trick”: constrain the attention mechanism to select only few frames
 - Keep up to K with highest scores
 - Limit selection to the vicinity of previous one

Decoding With Language Models

- Extend the beam search cost

$$\hat{Y} = \arg \min_Y -\log p_{\Theta}(Y|X) - \alpha p_{LM}(Y)$$

Transcript	LM cost $\log p(y)$	Model cost $\log p(y x)$	
"chase is nigeria's registrar and the society is an independent organization hired to count votes"	-108.5	-34.5	Ground truth
"in the society is an independent organization hired to count votes"	-64.6	-19.9	Decoded
"chase is nigeria's registrar"	-40.6	-31.2	Severe Transcript Truncation
"chase's nature is register"	-37.8	-20.3	
""	-3.5	-12.5	

Promoting long transcripts

Seems easy:

$$\hat{Y} = \arg \min_Y -\log p_{\Theta}(Y|X) - \alpha p_{LM}(Y) - \beta |Y|$$

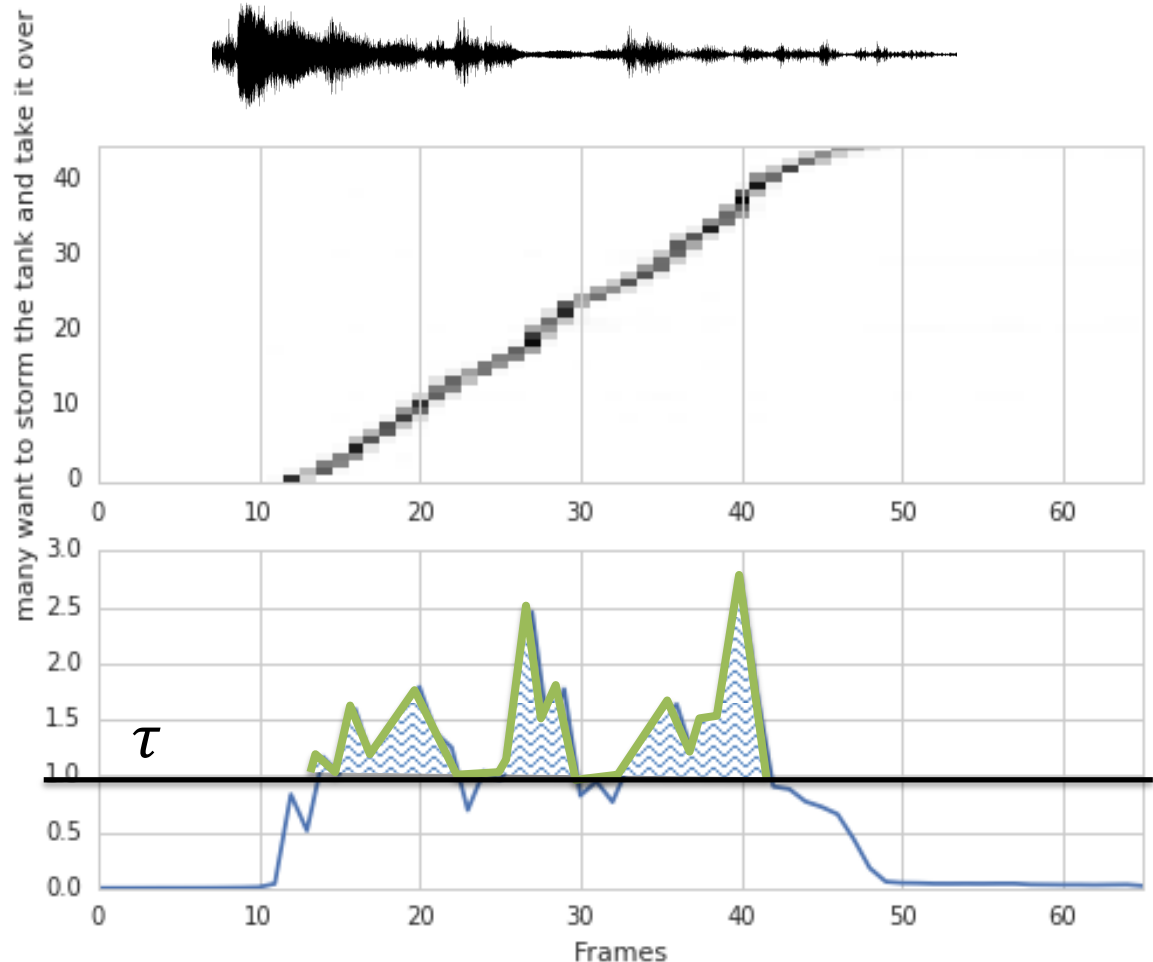
Problem: if any sequence of characters is cheap and the cost becomes negative, the model will keep repeating itself...

Coverage Criterion

Force decoding of all frames, but prevent looping.

$$\text{coverage} = \sum_f [\sum_i \alpha_{fi} > \tau]$$

Can't loop: a frame is counted at most once

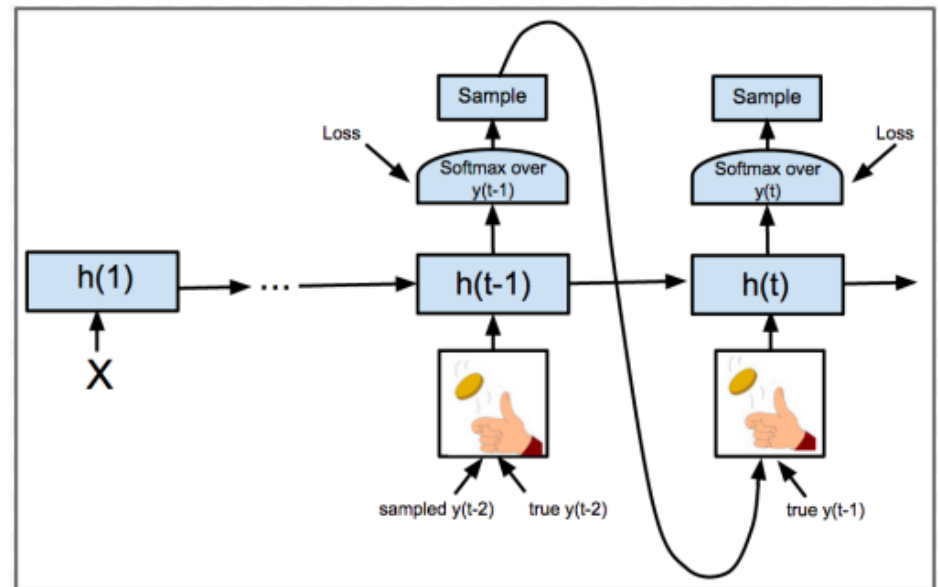


Better Training: Scheduled sampling

The problem:

- During training network sees no mistakes
- At test time the network makes mistakes

Pragmatic solution:
Allow mistakes
at training time.



Minimum Error Rate Training

- Augment cross-entropy loss

$$\sum_{i=1}^N \log p_{\Theta}(Y_i | Y_{<i}, X_i)$$

- With an expected error rate term:

$$\begin{aligned} \mathcal{L}_{\text{embr}} &= \mathbb{E}_{P(Y|X)}[\text{WER}(Y, Y^*)] \approx \\ &\approx \frac{1}{N} \sum_{Y \in \text{NBest}} \text{WER}(Y, Y^*) \hat{P}(Y|X) \end{aligned}$$

$$\text{where } \hat{P}(Y|X) = \frac{p_{\Theta}(Y|X)}{\sum_{Y' \in \text{NBest}} p_{\Theta}(Y'|X)}$$

A SOTA Attention ASR System

Seq2seq Voice Search reach state-of-the-art!

- Key ingredients:

- Lots of data: 12500 hours of audio
- Model changes: Multiple parallel attentions.
- Input representation: subword units (wordpieces). Slightly improve performance, greatly improve decoding speed! Less text normalization needed than for phonemes.
- Regularization: Label Smoothing, Scheduled Sampling.
- Minimum error-rate training.
- External language model integration (implemented as n-best list rescoring).

STATE-OF-THE-ART SPEECH RECOGNITION WITH SEQUENCE-TO-SEQUENCE MODELS, Chung-Cheng Chiu, Tara N. Sainath, Yonghui Wu, Rohit Prabhavalkar, Patrick Nguyen, Zhifeng Chen, Anjuli Kannan, Ron J. Weiss, Kanishka Rao, Katya Gonina, Navdeep Jaitly, Bo Li, Jan Chorowski, Michiel Bacchiani

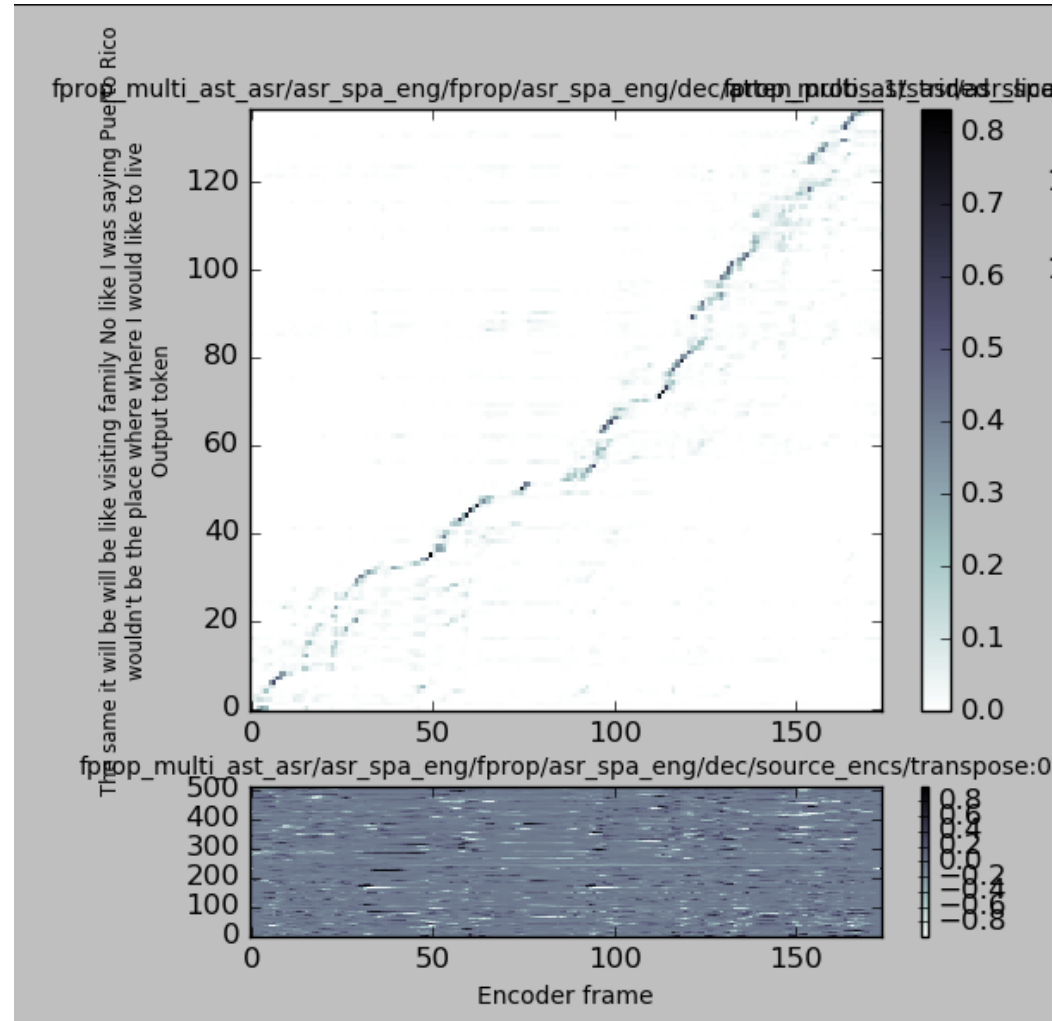
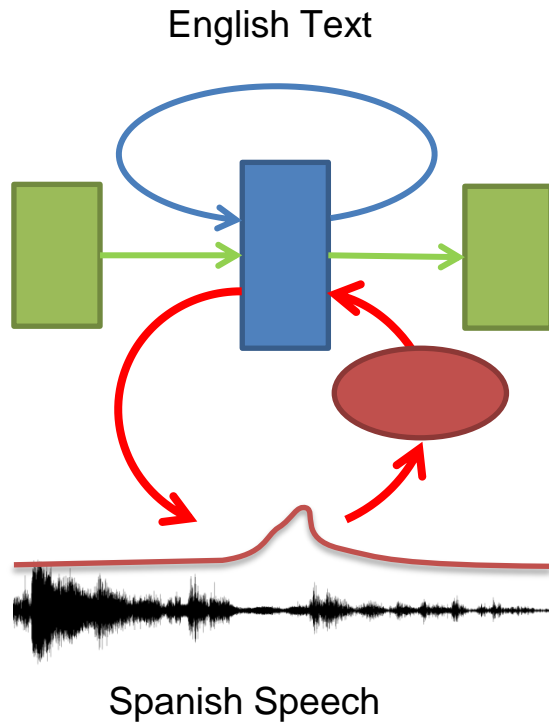
Other Examples of End-to-end Systems

Speech Translation

- Task: **Spanish speech** to English text translation
 - Typically train specialized translation model on ASR output lattice, or integrate ASR and translation decoding using e.g. stochastic FST
- Why end-to-end?
 - Directly optimize for desired output, avoid compounding errors
 - e.g. difficult for text translation system to recover from gross misrecognition
 - Single decoding step -> low latency inference
 - Less training data required -- don't need both transcript *and* translations
 - (might not be an advantage)
- Use sequence-to-sequence neural network model
 - Flexible framework, easily admits multi-task training
 - Previous work
 - [Bérard et al, 2016] trained "Listen and Translate" seq2seq model on *synthetic speech*
 - [Duong et al, 2016] seq2seq model to *align* speech with translation

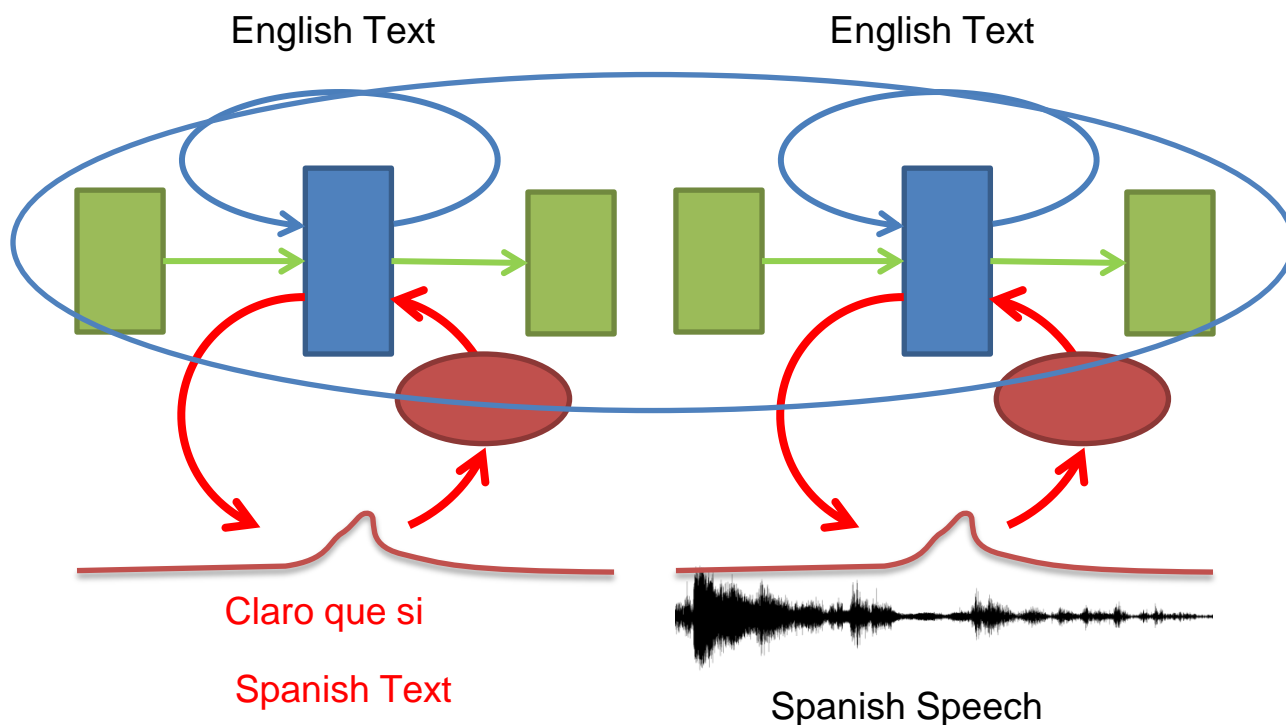
Our approach

- Seq2seq model



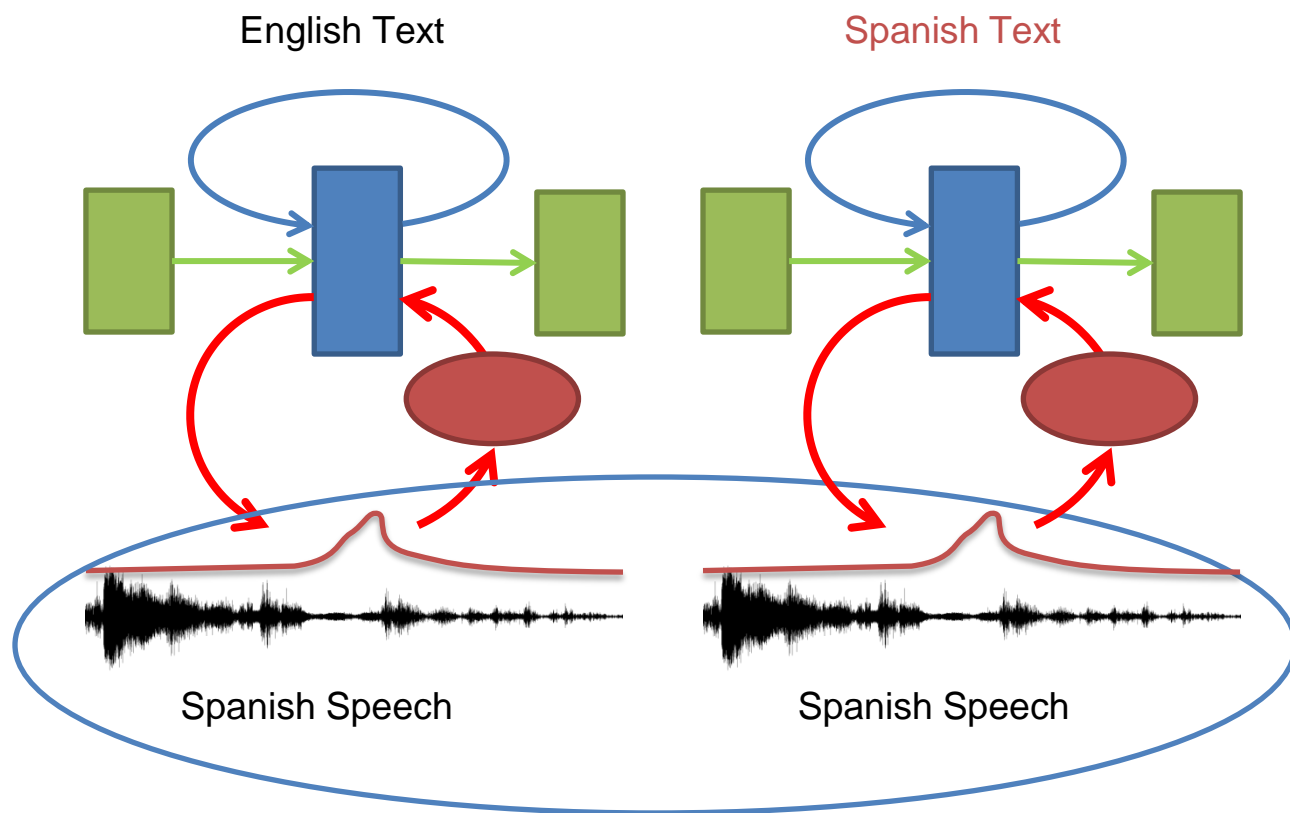
Multitask Learning, or Exploit All Data

Share weights of the decoder, separate encoders

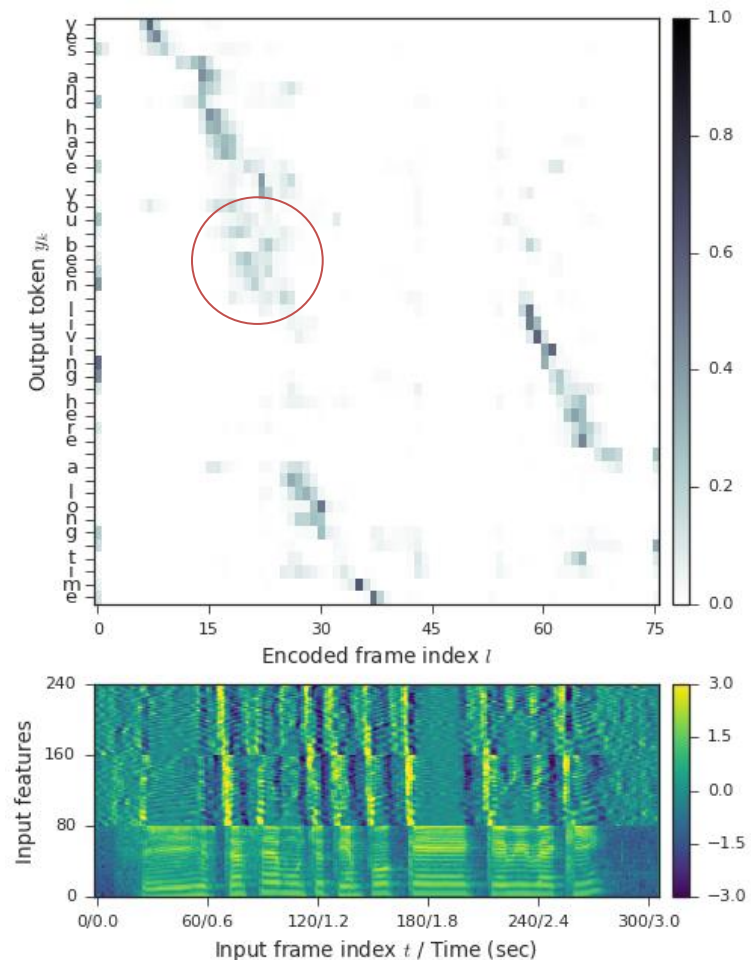
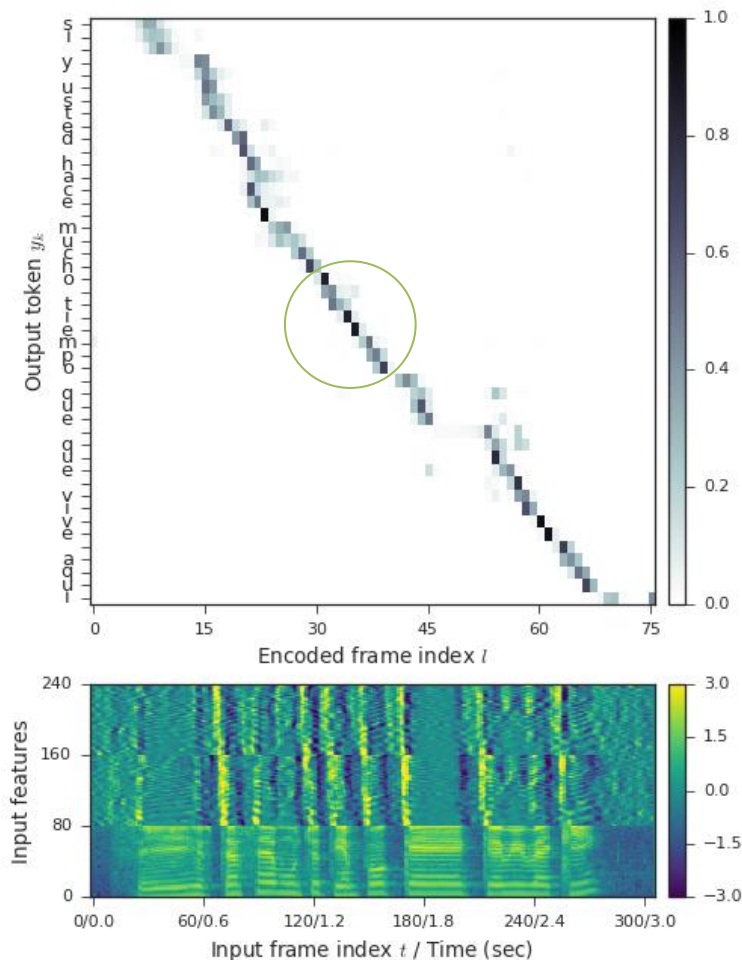


Multitask Learning, or Exploit All Data

Share weights of the encoder, separate decoders

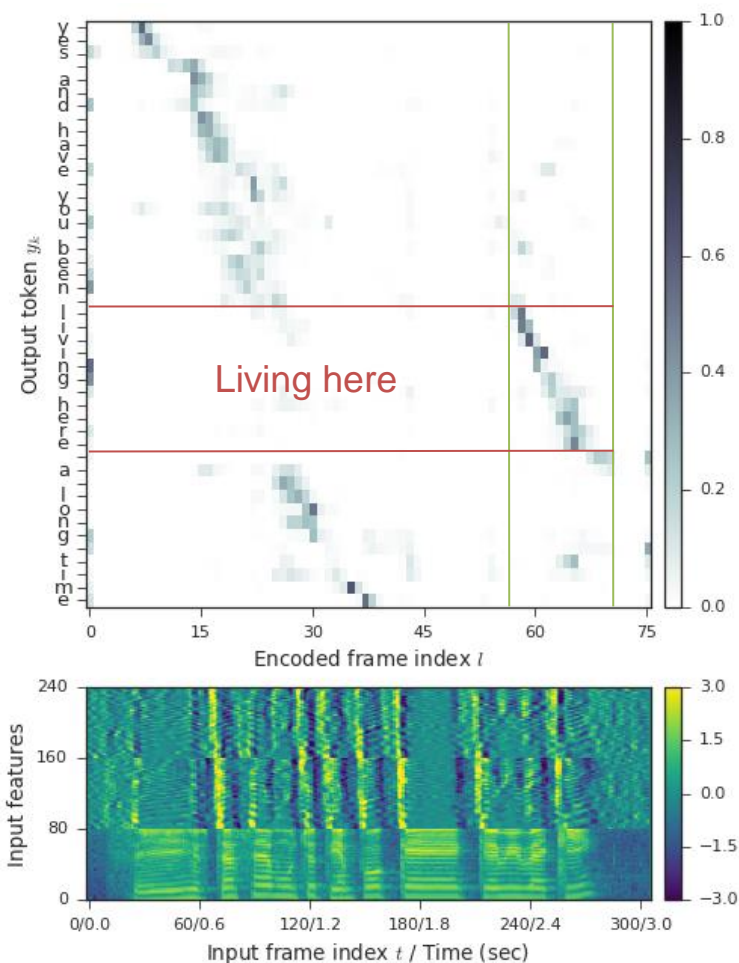
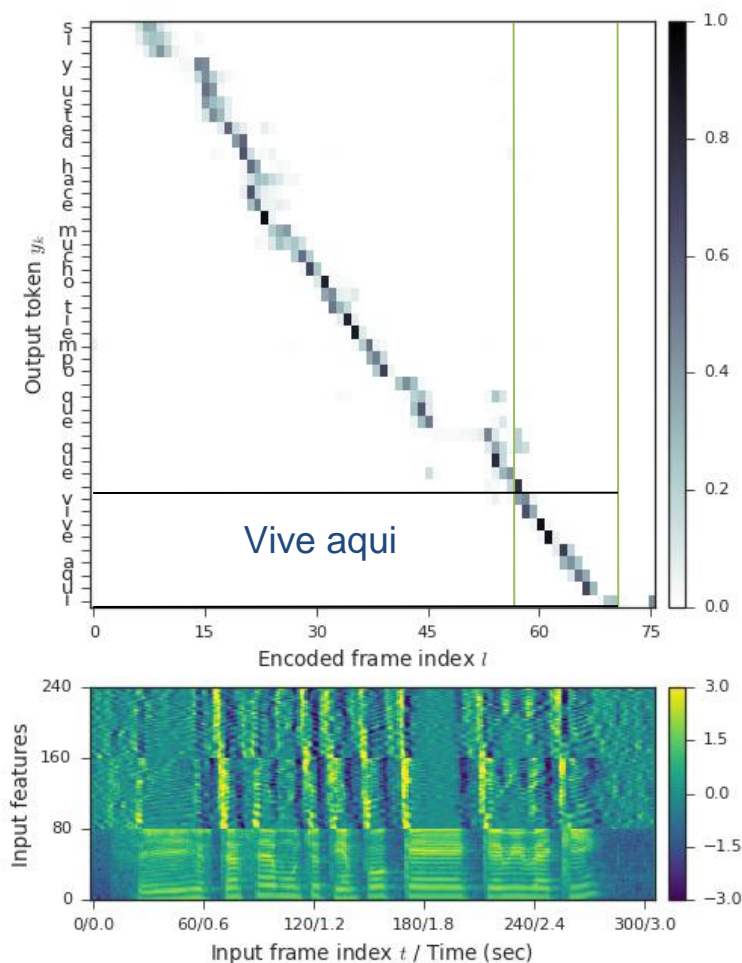


Seq2seq Speech Translation: Attention



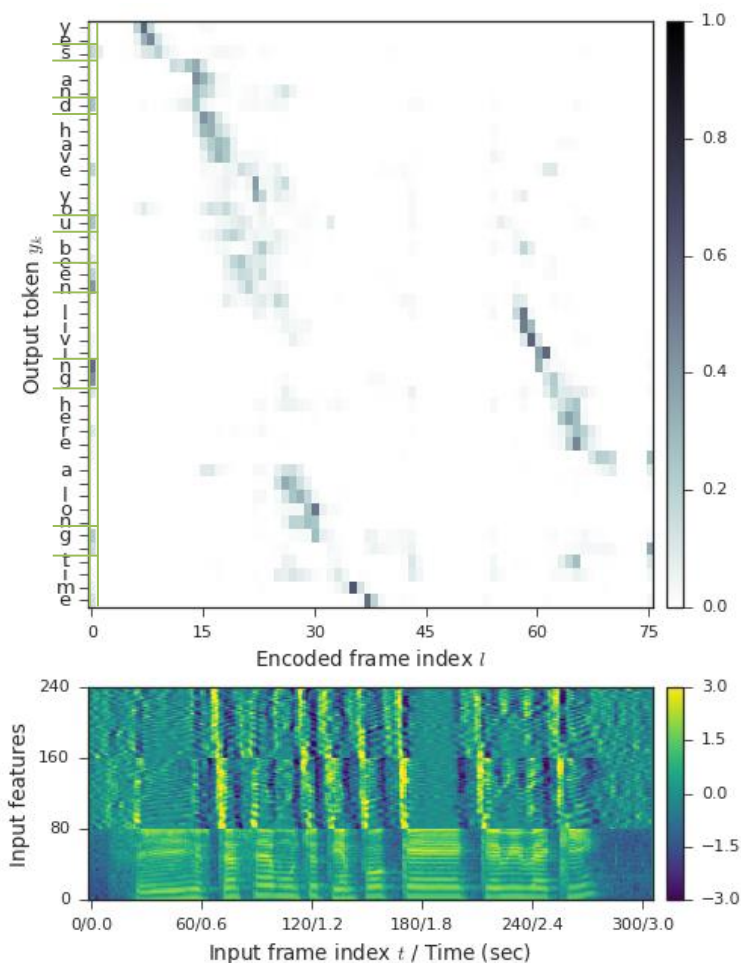
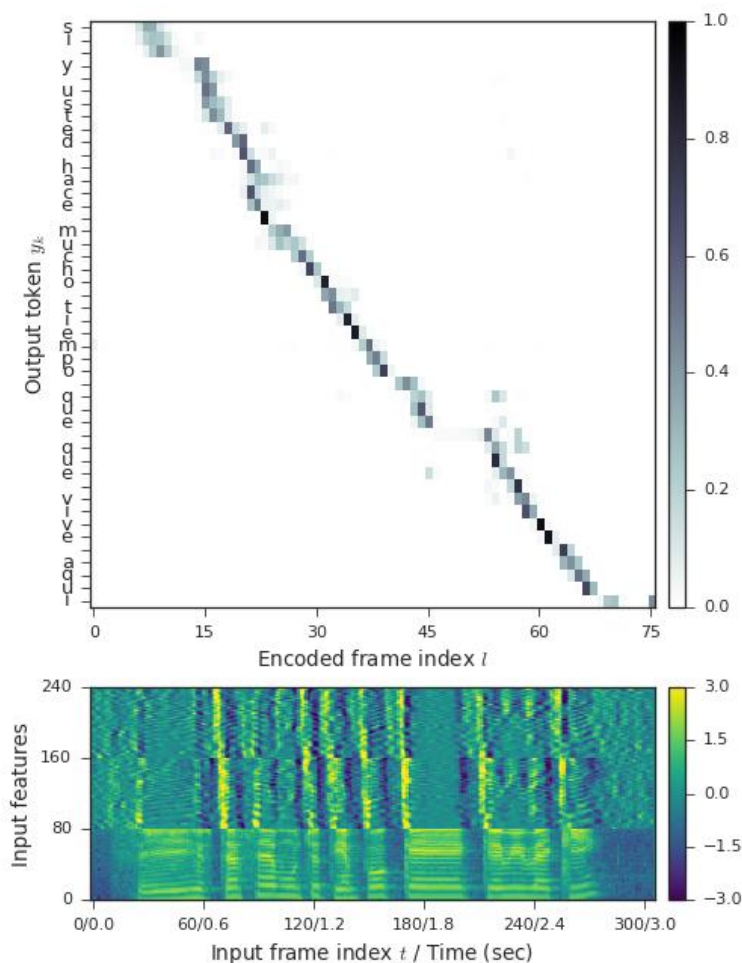
- recognition attention very **confident**
- translation attention **smoothed** out across many spectrogram frames for each output character
 - ambiguous mapping between Spanish speech acoustics and English text

Seq2seq Speech Translation: Attention



- speech recognition attention is mostly monotonic
- translation attention reorders input: **same frames** attended to for "vive aqui" and "living here"

Seq2seq Speech Translation: Example attention



- translation model attends to the beginning of input (i.e. silence) for the last few letters in each word
- already made a decision about word to emit, just acts a language model to spell it out.

Experiments: Fisher/Callhome Spanish-English data

- Transcribed Spanish telephone conversations from LDC
 - **Fisher**: conversations between strangers
 - **Callhome**: conversations between friends and family. more informal and challenging
- Crowdsourced English translations of Spanish transcripts from [Post et al, 2013]
- Train on 140k Fisher utterances (160 hours)
- Tune using Fisher/dev
- Evaluate on held out Fisher/test set and Callhome

Experiments: Baseline models

- **WER** on Spanish ASR

- seq2seq model outperforms classical GMM-HMM [19] and DNN-HMM [21] baselines

	dev	Fisher dev2	test	Callhome devtest	evltest
Ours ³	25.7	25.1	23.2	44.5	45.3
Post et al. [19]	41.3	40.0	36.5	64.7	65.3
Kumar et al. [21]	29.8	29.8	25.3	–	–

- **BLEU score** on Spanish-to-English text translation

- seq2seq NMT (following [Wu et al, 2016]) slightly underperforms phrase-based SMT baselines

	dev	Fisher dev2	test	Callhome devtest	evltest
Ours	58.7	59.9	57.9	28.2	27.9
Post et al. [19]	–	–	58.7	–	27.8
Kumar et al. [21]	–	65.4	62.9	–	–

Experiments: End-to-end speech translation

Model	dev	Fisher		Callhome	
		dev2	test	devtest	evltest
End-to-end ST ³	46.5	47.3	47.3	16.4	16.6
Multi-task ST / ASR ³	48.3	49.1	48.7	16.8	17.4
ASR→NMT cascade ³	45.1	46.1	45.5	16.2	16.6
Post et al. [19]	—	35.4	—	—	11.7
Kumar et al. [21]	—	40.1	40.4	—	—

- BLEU score (higher is better)
- Multi-task > End-to-end ST > Cascade >> non-seq2seq baselines
- ASGD training with 10 replicas (16 for multitask)
 - ASR model converges after 4 days
 - ST and multi-task models continue to improve for 2 weeks

Example output: compounding errors

ASR

ref: "sí a mime gusta mucho bailar merengue y **salsa** también"

hyp: "sea me gusta mucho bailar merengue y **sabes** también"

hyp: "sea me gusta mucho bailar medio **inglés**"

hyp: "o sea me gusta mucho bailar merengue y **sabes** también"

hyp: "sea me gusta mucho bailar medio **inglés** **sabes** también"

hyp: "sea me gusta mucho bailar merengue"

hyp: "o sea me gusta mucho bailar medio **inglés**"

hyp: "sea no gusta mucho bailar medio **inglés**"

hyp: "o sea me gusta mucho bailar medio **inglés** **sabes** también"

End-to-end ST

ref: "yes i do enjoy dancing merengue and **salsa** music too"

hyp: "i really like to dance merengue and **salsa** also"

hyp: "i like to dance merengue and salsa also"

hyp: "i don't like to dance merengue and salsa also"

hyp: "i really like to dance merengue and salsa and also"

hyp: "i really like to dance merengue and salsa"

hyp: "i like to dance merengue and salsa and also"

hyp: "i like to dance merengue and salsa"

hyp: "i don't like to dance merengue and salsa and also"

Cascade: ASR top hypothesis -> NMT

hyp: "i really like to dance merengue and **you know** also"

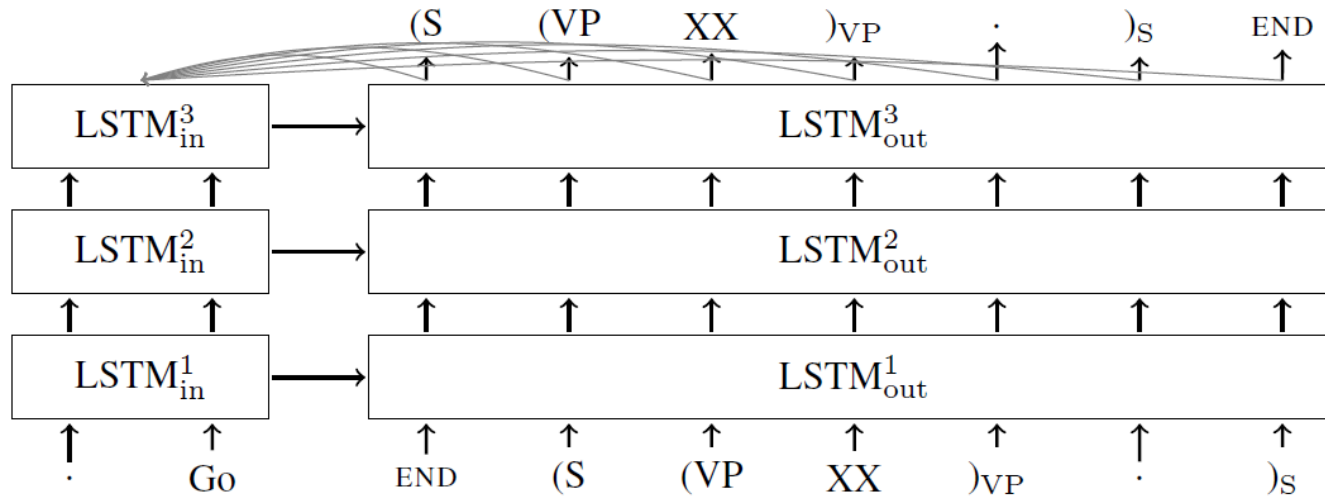
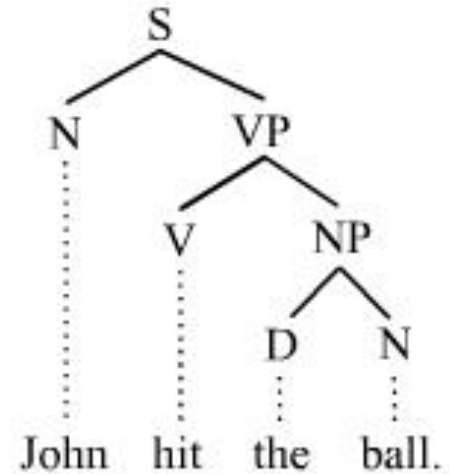
- ASR consistently mis-recognizes "merengue y salsa" as "merengue y **sabes**" or "medio **inglés**"
- NMT has no way to recover

End-to-end systems in NLP:

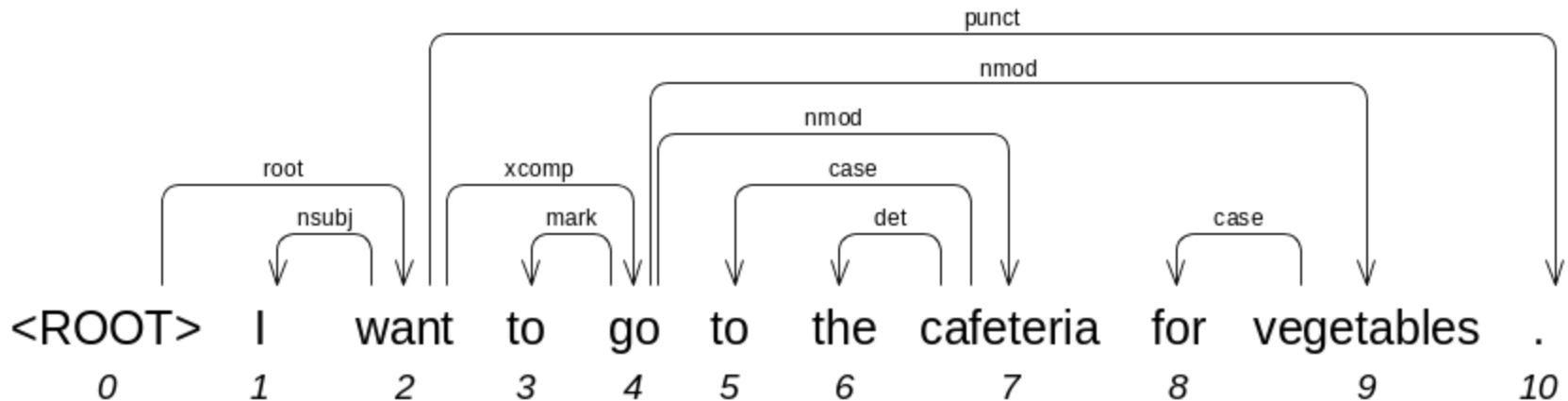
How to parse sentences?

For constituency parsing:
Treat parsing as a sequence-to-sequence problem:

- Input: sentence
„Go .”
- Output: linearized parse tree:
„(S (VP XX)VP .)S END”



Dependency parsing

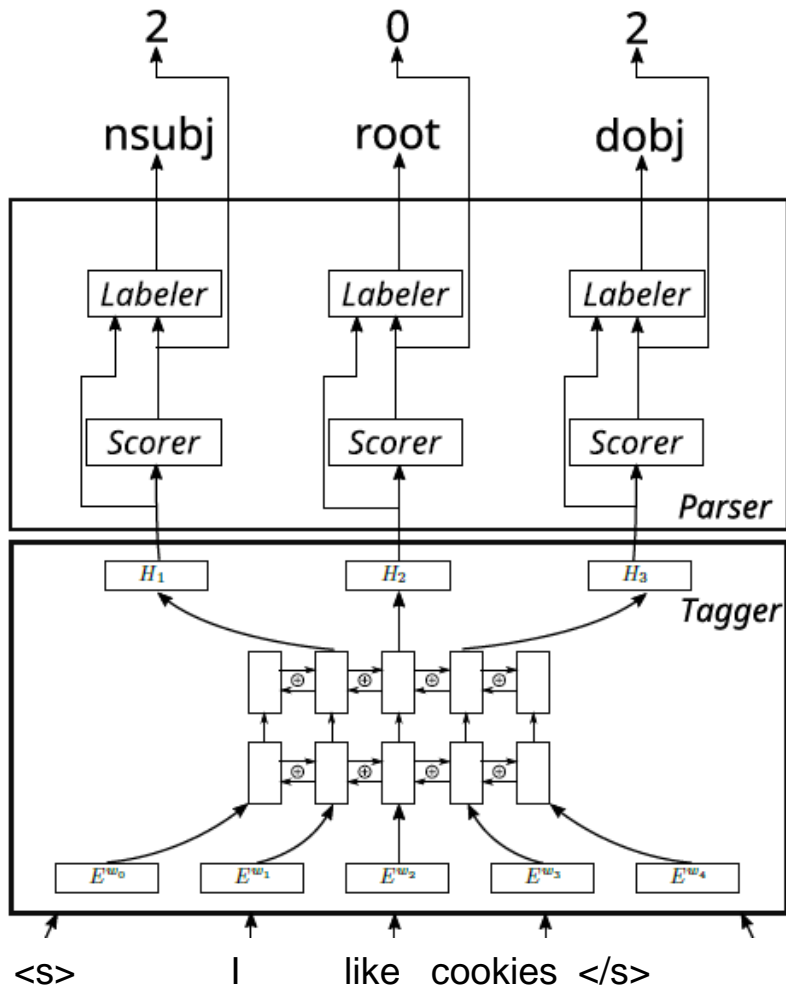


- Desired output: directed edges between words.
- At each step the attention selects a few words.
- Idea: use the selection weights as pointers.

Chorowski et al. "Read, Tag, and Parse All at Once, or Fully-neural Dependency Parsing",
arxiv <https://arxiv.org/pdf/1609.03441>

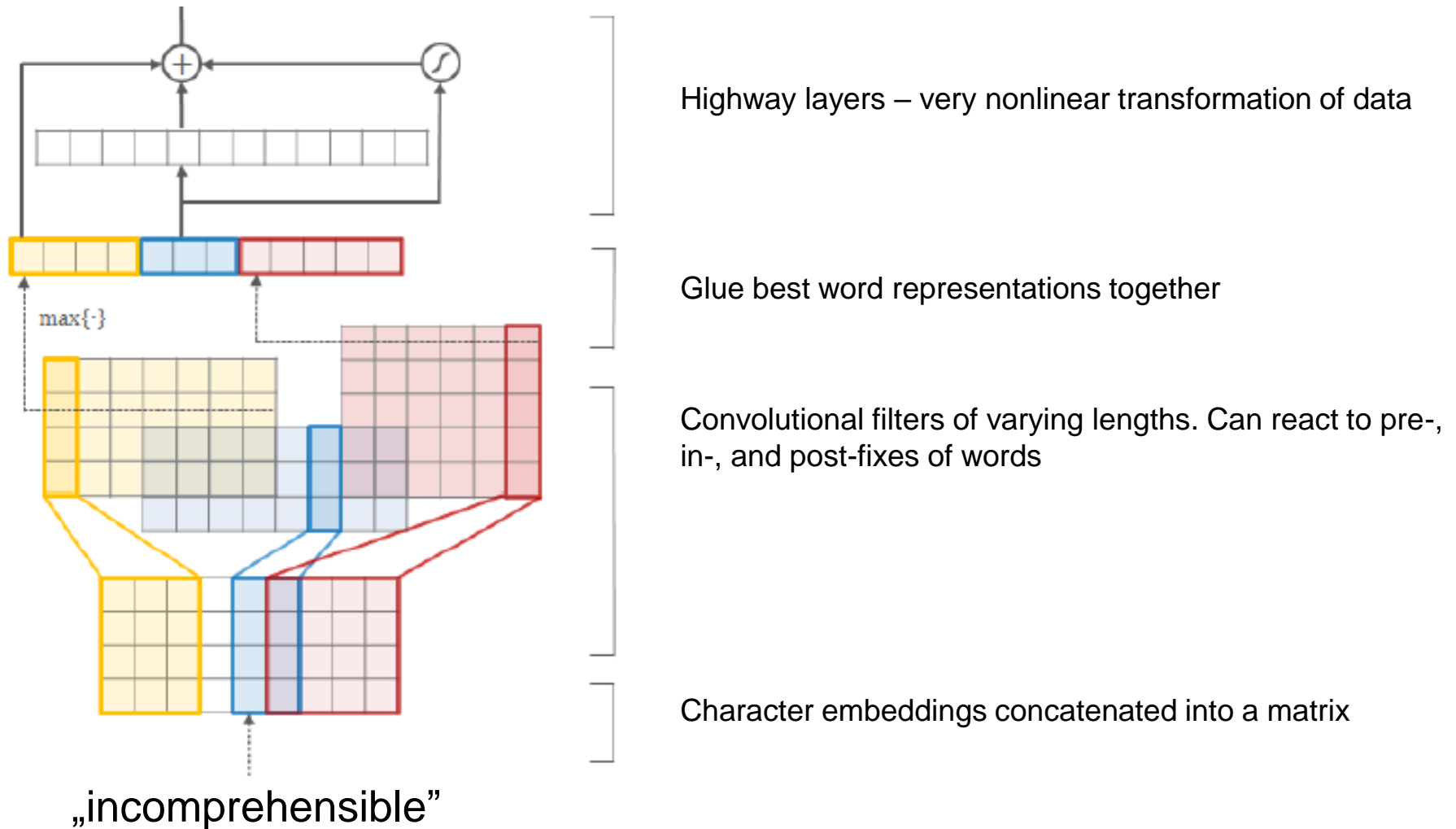
Zapotoczny et al. "On Multilingual Training of Neural Dependency Parsers" TSD 2017

Dependency parsing

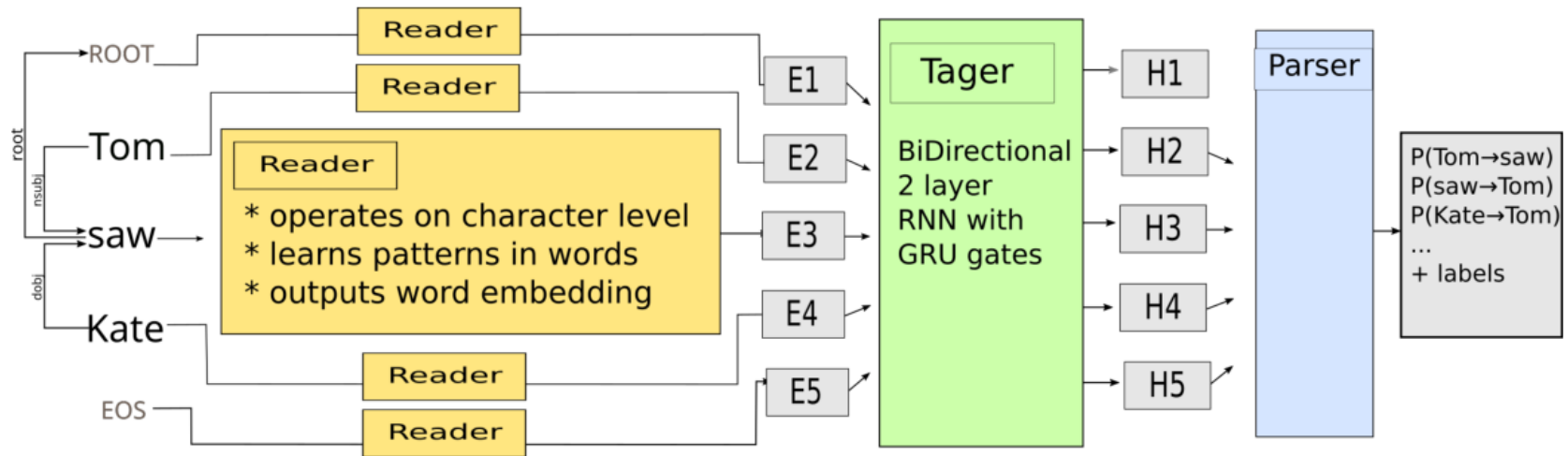


- For each word w
- Two operations:
1. Find head h (use attention mechanism)
 2. Use (w, h) to predict dependency type

From characters to word embeddings



From characters to parse trees



Reader reads orthographic representations of words and is sensitive to morphemes.

Tagger puts words into context

Parser finds the dependency edges.

Multitask Learning is King

Polish language has small number of dependency parsing examples (8 227 sentences, 83 571 words)

- Use related data for auxiliary costs:
 - The *tagger* subnetwork predicts POS tags
- Multilingual training is possible:
 - Polish and Czech are similar.
 - Czech language has much better dataset (77 765 sentences, 1332 566 words)

Parser results

Results on common dependencies treebanks

language	#sentences	Ours		SyntaxNet		ParseySaurus	
		UAS	LAS	UAS	LAS	UAS	LAS
Czech	87 913	91.41	88.18	89.47	85.93	89.09	84.99
Polish	8 227	90.26	85.32	88.30	82.71	91.86	87.49
Russian	5 030	83.29	79.22	81.75	77.71	84.27	80.65
German	15 892	82.67	76.51	79.73	74.07	84.12	79.05
English	16 622	87.44	83.94	84.79	80.38	87.86	84.45
French	16 448	87.25	83.50	84.68	81.05	86.61	83.1
Ancient Greek	25 251	78.96	72.36	68.98	62.07	73.85	68.1

We can share data from multiple languages:

Shared parts	Main lang	Aux lang	UAS	LAS
-	Polish	-	90.31	85.21
<i>Parser</i>	Polish	Czech	90.72	85.57
<i>Tagger, Parser</i>	Polish	Czech	91.19	86.37
<i>Tagger, POS Predictor, Parser</i>	Polish	Czech	91.65	86.88
<i>Reader, Tagger, POS Predictor, Parser</i>	Polish	Czech	91.91	87.77

<http://arxiv.org/abs/1609.03441>, <http://arxiv.org/abs/1705.10209>

Jabberwocky (Lewis Carroll)

Tw'as brillig and the slithy toves

Did gyre and gimble in the wabe;

All mimsy were the borogoves,

And the mome raths outgrabe.

Żabrołak (Stanisław Barańczak)

Brzdęśniało już ślimonne prztowie
praet:sg:n:perf qub adj:sg:nom:n:pos subst:sg:nom:n

Wyrło i warło się w gulbieży
praet:sg:n:perf conj praet:sg:n:imperf qub prep:acc:nwok subst:pl:acc:m3

Zmimszałe ćwiły borogowie
adj:pl:acc:m3:pos praet:pl:f:imperf subst:pl:nom:m1

I rcie grdypały z mrzerzy
conj subst:pl:nom:n praet:pl:f:imperf prep:gen:nwok subst:sg:gen:f

Underlined words are neologisms, green are correct!

Multilingual Grammatical Relations

Polish word	Closest russian embeddings
przedwrześniowej	адренергической тренерской таврической непосредственной археологической философской <i>верхнюю</i>
większych	автомобильных <i>трёхдневные</i> технических практических официальных оригинальных
policyjnym	главным историческим глазным непосредственным <i>косыми</i> летним двухсимвольным

- **Green Russian** words have similar grammatical function to **Polish words**.
- **-ской** (skoy) and **-нной** (nnoy) quite distant from polish — **owej** (ovey).
- 3-letter **-ych** paired with 2 letter **-ых**

Is End-to-end Software 2.0?

One homogenous model vs large pipeline of many models.

All parts conspire to better work together.

Homogenous computation: low precision matrix multiplication

Axis of improvement:

Accuracy: more data, larger model

Speed: smaller model, less precise computations

Questions:

Unit tests for parts of the model?

What if there is little or no data?

How to maintain it over time?

How to adapt to special cases?

References & Further Reading

- D. Amodei, et al., “Deep Speech 2: End-to-End Speech Recognition in English and Mandarin,” *arXiv:1512.02595 [cs]*, Dec. 2015.
- Luo, Y., Chiu C, Jaitly N., Sutskever I., „Learning Online Alignments with Continuous Rewards Policy Gradient”, <https://arxiv.org/abs/1608.01281>
- Chiu, C. C., Sainath, T. N., Wu, Y., Prabhavalkar, R., Nguyen, P., Chen, Z., ... & Jaitly, N. (2017). State-of-the-art speech recognition with sequence-to-sequence models. *arXiv preprint arXiv:1712.01769*.
- Chorowski J., Zapotoczny M., Rychlikowski P., „Read, Tag, and Parse All at Once, or Fully-neural Dependency Parsing”, <http://arxiv.org/abs/1609.03441>
- Chorowski J., Bahdanau, Serdyuk, D., D., Cho, K. Bengio, Y., Attention-Based Models for Speech Recognition, NIPS 2015
- Chorowski J., Bahdanau, D., Cho, K. Bengio, Y., End-to-end Continuous Speech Recognition using Attention-based Recurrent NN: First Results, Deep Learning Workshop at NIPS 2014
- Jan Chorowski, Navdeep Jaitly, Towards better decoding and language model integration in sequence to sequence models, *arXiv:1612.02695*
- Yu Zhang, William Chan, Navdeep Jaitly, Very Deep Convolutional Networks for End-to-End Speech Recognition, *arXiv:1610.03022*
- Bahdanau D., Serdyuk D., Brakel P., Ke N., Chorowski J., Courville A., Bengio Y., „TASK LOSS ESTIMATION FOR SEQUENCE PREDICTION”, <http://arxiv.org/abs/1511.06456>
- Bahdanau, D., Chorowski, J., Serdyuk, D., Brakel, P., Bengio, Y., End-to-End Attention-based Large Vocabulary Speech Recognition, *arXiv:1508.04395*
- Bahdanau, D., Cho, K., and Bengio, Y. (2014). Neural machine translation by jointly learning to align and translate. ICLR2015 *arXiv:1409.0473*
- N. Jaitly et al, “A Neural Transducer”, NIPS 2016
- Johnson, Melvin, et al. "Google's multilingual neural machine translation system: enabling zero-shot translation." *arXiv preprint arXiv:1611.04558* (2016).
- Norouzi et al, “Reward Augmented Maximum Likelihood for Neural Structured Prediction”, NIPS 2016
- W. Chan, N. Jaitly, Q. V. Le, and O. Vinyals, “Listen, Attend and Spell,” *arXiv:1508.01211 [cs, stat]*, Aug. 2015.
- A. Graves, “Practical Variational Inference for Neural Networks,” in *Advances in Neural Information Processing Systems 24*, J. Shawe-Taylor, R. S. Zemel, P. L. Bartlett, F. Pereira, and K. Q. Weinberger, Eds. Curran Associates, Inc., 2011, pp. 2348–2356.
- Graves, A. (2013). Generating sequences with recurrent neural networks. *arXiv:1308.0850*
- Graves, A., Mohamed, A.R., and Hinton, G. (2013b). Speech recognition with deep recurrent neural networks. IEEE ICASSP 2013
- Sutskever, I., Vinyals, O., Le, Quoc, Sequence to Sequence Learning with Neural Networks, NIPS 2014
- A. Vaswani, Ashish, et al. "Attention Is All You Need." *arXiv preprint arXiv:1706.03762* (2017).
- Vinyals O., Toshev A., Bengio, S., Erhan, D. „Show and Tell: A Neural Image Caption Generator”, <https://arxiv.org/abs/1411.4555>
- Vinyals, O., Kaiser, L., Koo, T., Petrov, S., Sutskever, I., Hinton, G., Grammar as a Foreign Language, NIPS 2015
- Vinyals, O., Fortunato, M., Jaitly, N., Pointer Networks, NIPS 2015
- Weiss, R. J., Chorowski, J., Jaitly, N., Wu, Y., & Chen, Z. (2017). Sequence-to-sequence models can directly translate foreign speech. In *Proc. Interspeech*.