Uniwersytet Wrocławski

Google

End-to-end approaches to speech recognition and language processing

Jan Chorowski
University of Wroclaw & 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 Zapotoczny
- Paweł Rychlikowski
- William Chan
- Navdeep Jaitly

Outline

- End-to-end speech recognition systems
- Challenges and promising research directions
- Uses of attention mechanism for NLP

Motivation: 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)
- present a 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

End-to-end systems are the future

Recent examples of end-to-end systems

- 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 (many recent papers)

Are DNN-HMMs end-to-end trainable?

Without sequence discriminative training: **no**

- Lexicon and HMM structure are not optimized with the rest of the system
- Acoustic model (DNN) is trained to predict the states of the HMM in isolation from the language model

With sequence discriminative training: **more end-to-end**, but still **no**

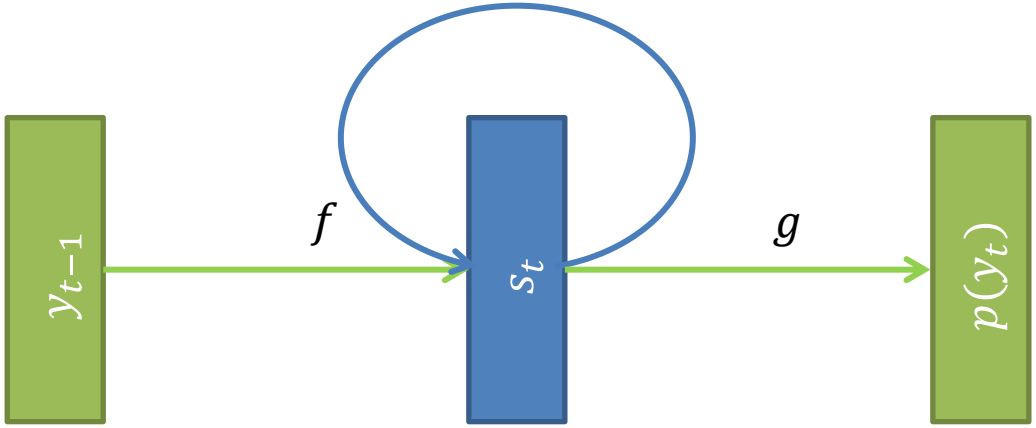
- Lexicon and HMM structure …

Our Goal

- Directly model $p(Y|X)$
  where Y: sequence of words or characters
  X: recording

- Compare with the classical decomposition
$$p(Y|X) \propto p(Y)p(X|Y)$$

RNNs Learn $p(Y)$



Decompose

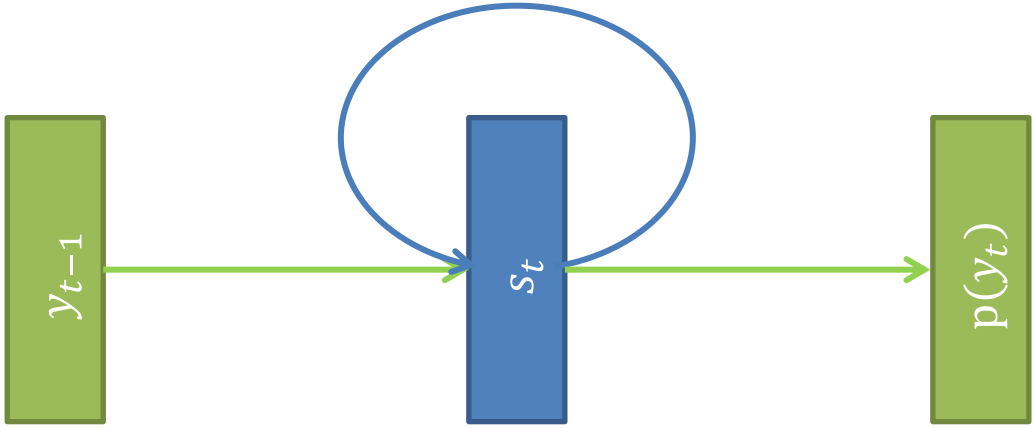
$$p(Y) = \prod p(y_t | y_{t-1}, y_{t-2}, \cdots, y_1)$$

Model the probabilities using a recurrent relation

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

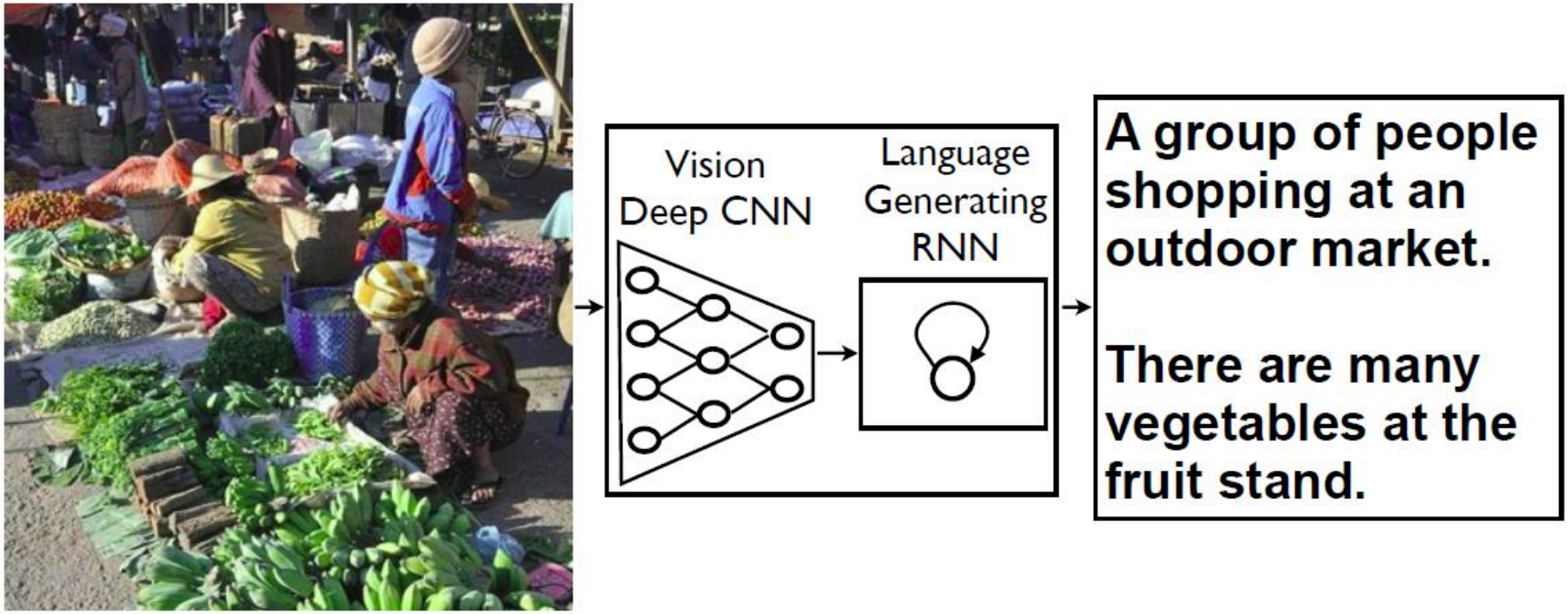
$$s_t = f(s_{t-1}, y_{t-1})$$

How to condition an RNN?



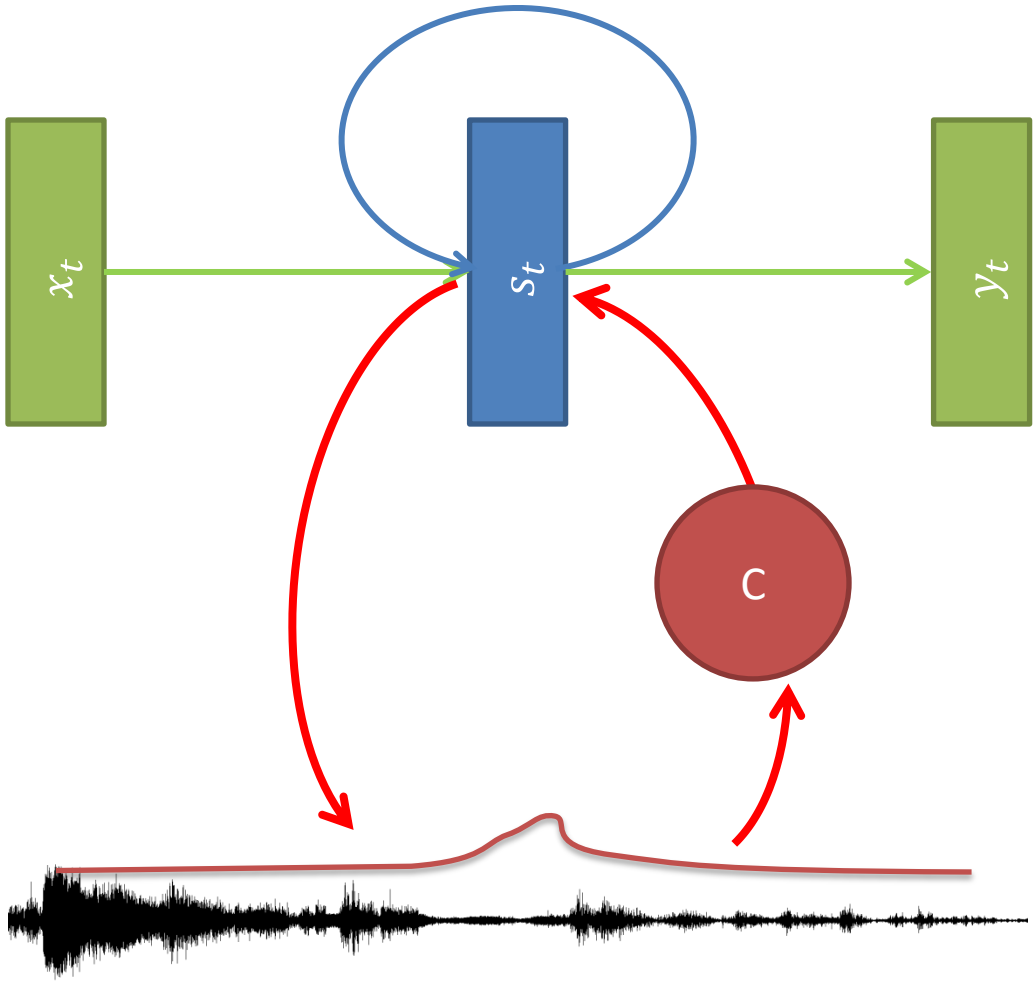
- Idea #1: conditioned through the first hidden state
- Idea #2: condition separately on every step

Idea #1
condition through the 1st hidden state



O. Vinyals et al, “Show and Tell: a Neural Image Caption Generator https://arxiv.org/abs/1411.4555

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

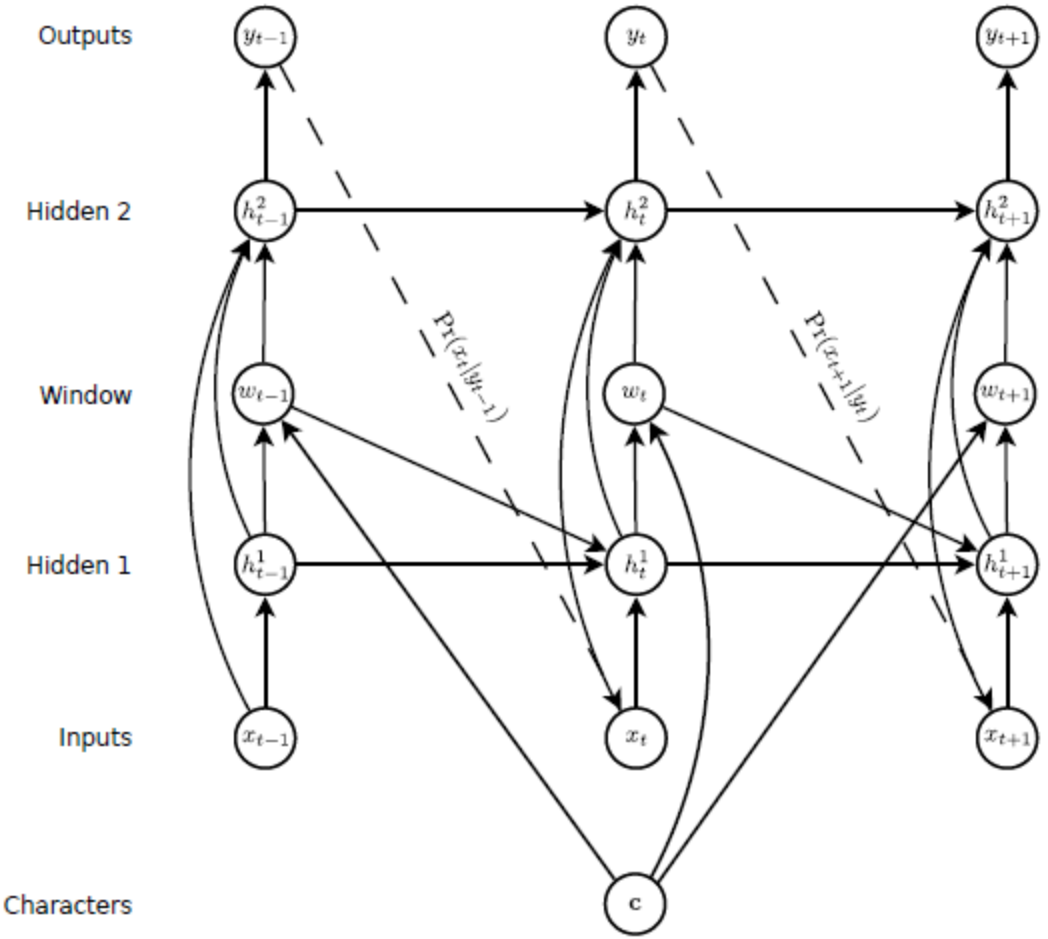
$$s_t = f(s_{t-1}, y_{t-1}, c)$$

Soft attention originally proposed for the Montreal neural translation project [Bahdanau2014]

Attention mechanism in RNNs

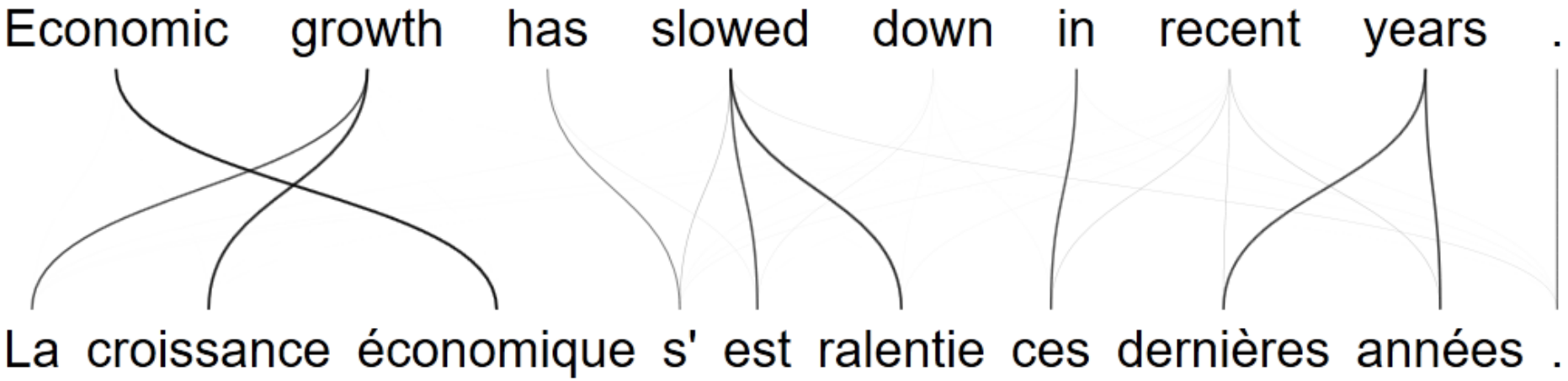
from his travels it might have been
from his travels it might have been
from his travels it might have been

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

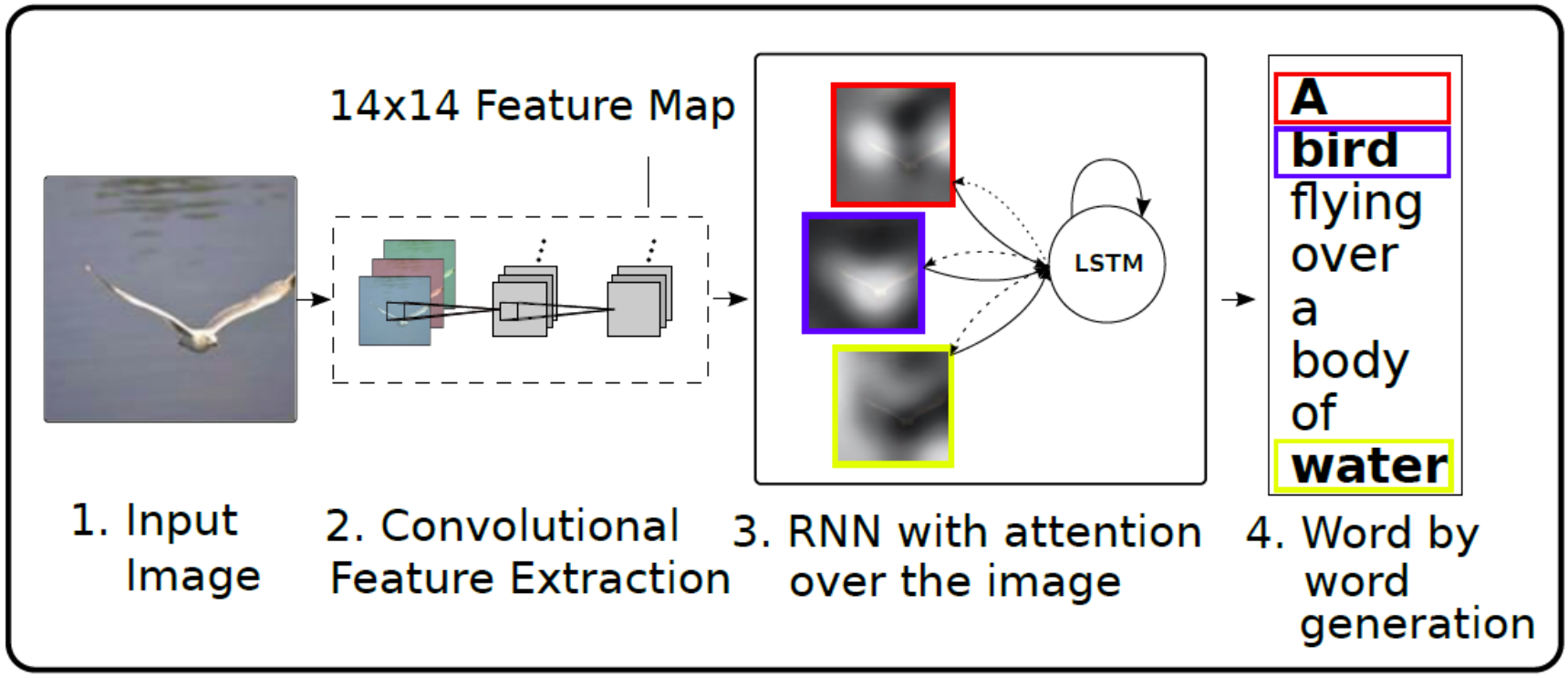


Graves, A., 2013. Generating Sequences With Recurrent Neural Networks. arXiv:1308.0850 [cs]

Attention mechanism in translation



Attention mechanism for captioning



http://arxiv.org/pdf/1502.03044.pdf

Our System at a Glance

Network defines

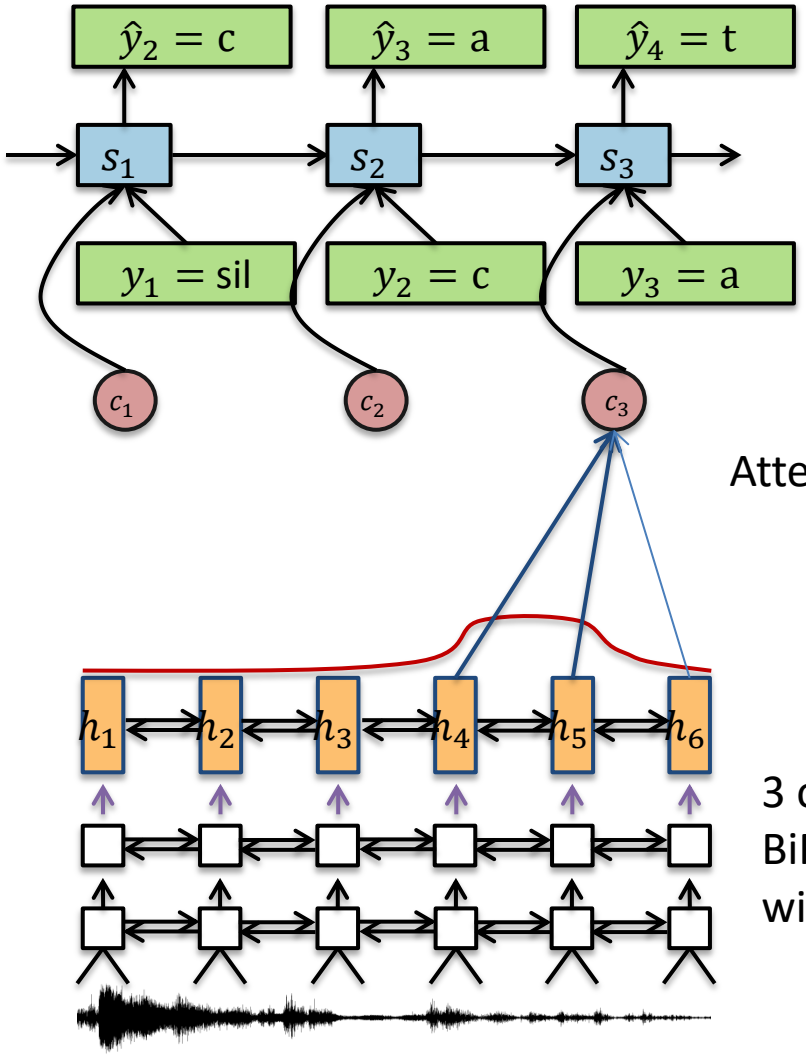
$$p_\Theta(Y|X)$$

where

$Y = y_1 y_2 \ldots y_T$ are characters

$X = x_1 x_2 \ldots x_F$ are speech frames

Θ are parameters



GRU RNN
with 250 units

Attention mechanism

3 or 4 GRU
BiRNN layers
with 250 units each

Training

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

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

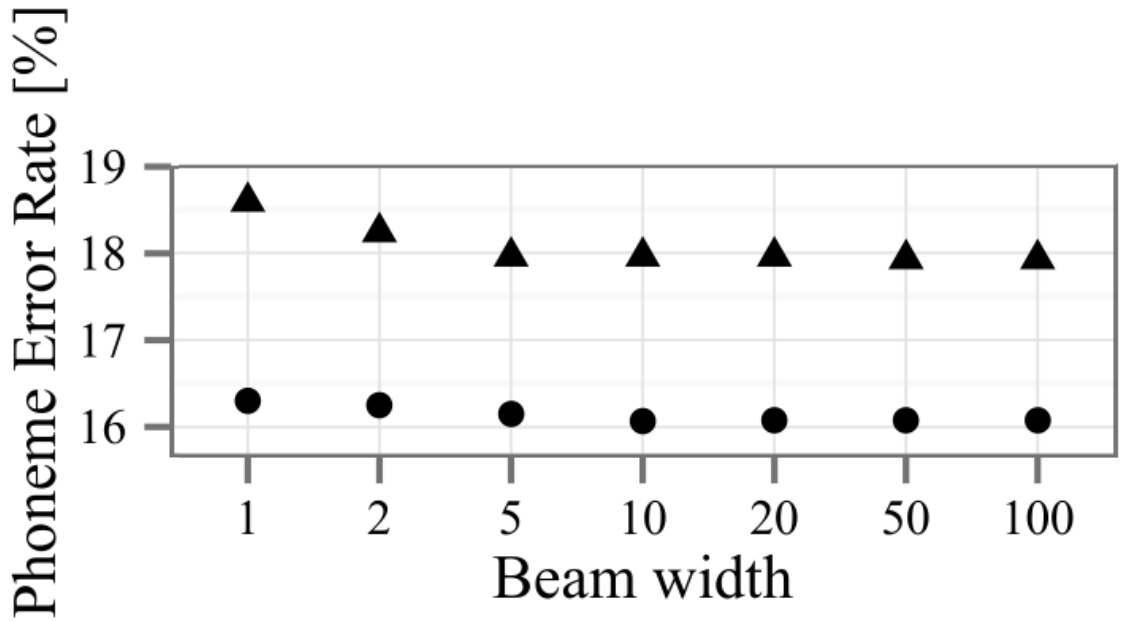
p_{Θ} is differentiable with respect to Θ thus we can use gradient based methods

Decoding

Use beam search to find

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

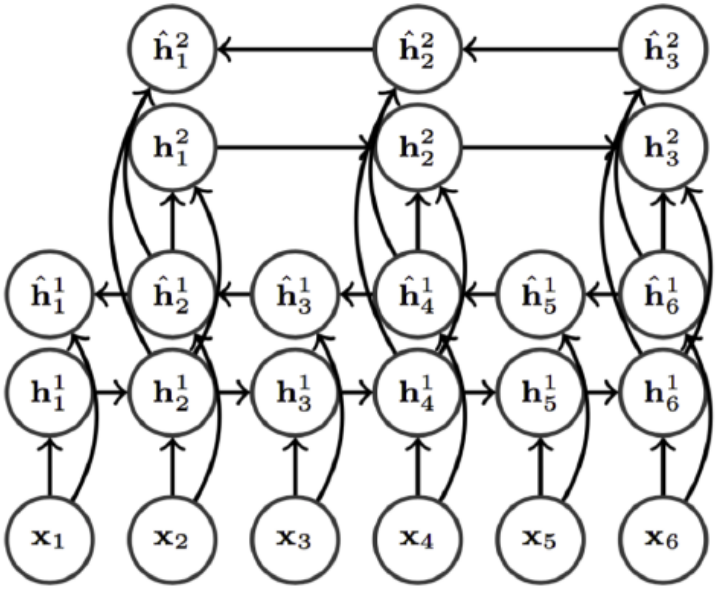
Narrow beams are needed:



Tricks of the Trade: 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: Regularization

- RNNs don’t like weight decay too much
- Constraining the norm of incoming weights to 1 for every unit of the network gave us ~30% performance improvement
- Weight noise was crucial on TIMIT!

Comparison with other Approaches

- The alignment α is explicitly computed

- Compare with alignment as a latent variable to be marginalized:

$$P_\Theta(Y|X) = \sum_\alpha P_\Theta(Y, \alpha|X)$$

Some Results

Results obtained on TIMIT

Model	Dev	Test
Content-based Attention	15.9%	18.7%
Location-aware attention	15.8%	17.6%
(Graves 2013) RNN Transducer	N/A	17.7%
(Toth 2014) HMM over Conv. Nets	13.9%	16.7%

Results obtained on WSJ (character based) (ICASSP2016)

Model	CER	WER
Attention-based model	6.7	18.9
Attention-based model + extended trigram LM	3.9	9.3
(Graves 2014) CTC + expected transcription loss	8.4	27.3
(Hannun 2014) CTC + bigram LM	5.7	14.1
(Miao 2015) CTC + extended trigram LM	N/A	7.3

More Results
Listen, Attend and Spell (Chan 2015)

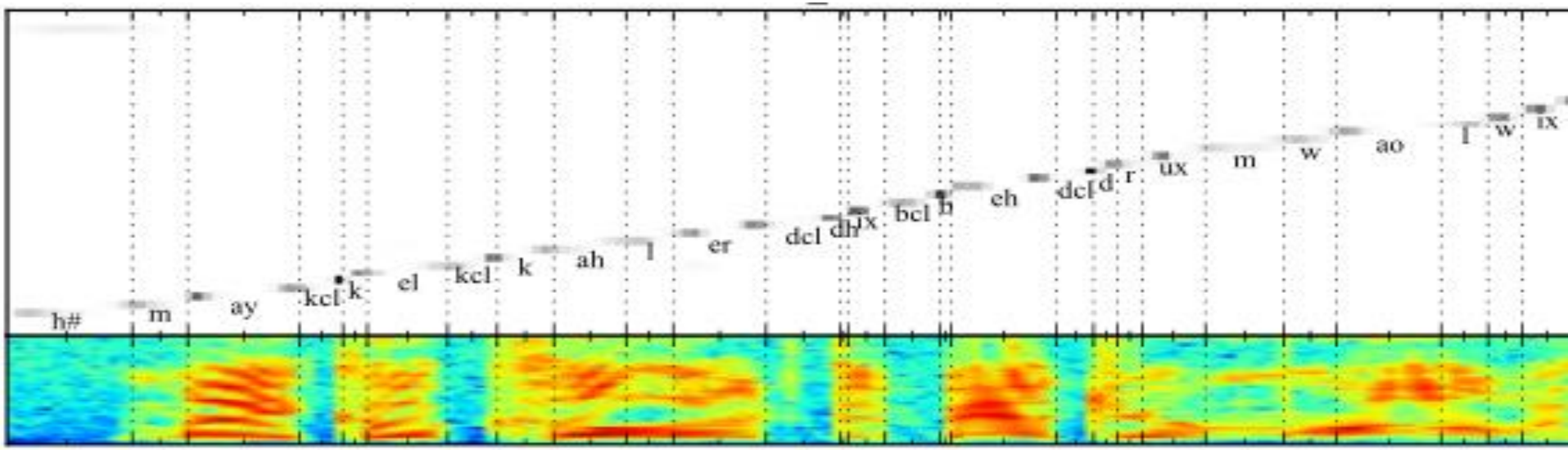
- Trained on Google data (2000h)
- Learned “AAA=Triple A”!

Model	Clean WER	Noisy WER
CLDNN-HMM [20]	10.9	11.9
LAS	16.2	19.0
LAS + LM Rescoring	12.6	14.7
LAS + Sampling	14.2	16.3
LAS + Sampling + LM Rescoring	11.2	12.8

Beam	Text	Log Probability	WER
Truth	call aaa roadside assistance	-	-
1	call aaa roadside assistance	-0.5740	0.00
2	call triple a roadside assistance	-1.5399	50.00
3	call trip way roadside assistance	-3.5012	50.00
4	call xxx roadside assistance	-4.4375	25.00

Challenges

- Dealing with long sequences and repetitions.
- What about text-only data?
- Is cross-entropy the best training objective?
- On-line decoding?



A simple experiment

Train a network on TIMIT

Concatenate test utterances a few times

Decode as usual

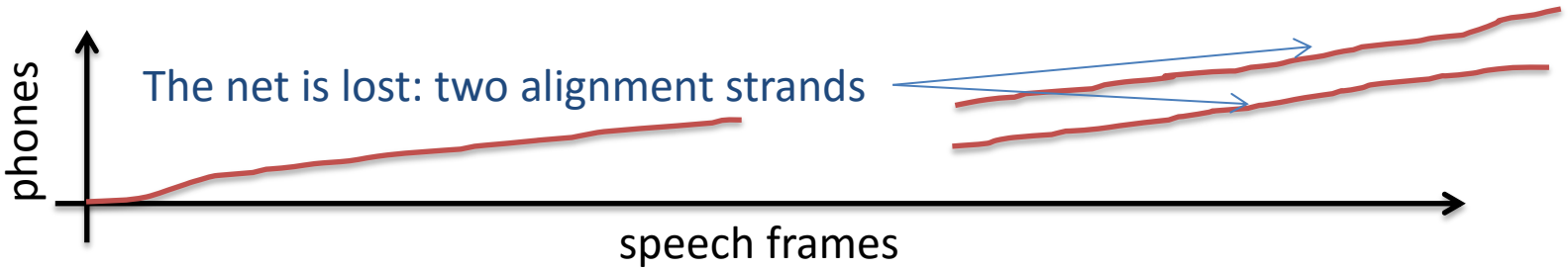
Performance drops dramatically
On long utterances decoding completely fails

Investigation of Long Inputs

The setup:

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

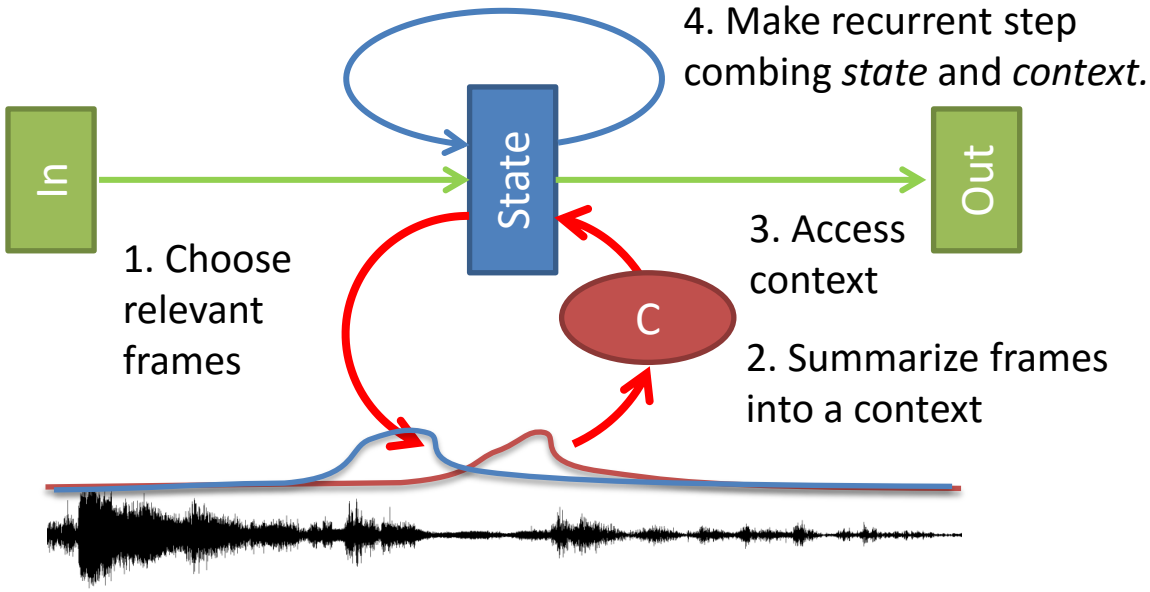
Typical result



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

Chorowski et al., „Attention-based models for speech recognition”, NIPS 2015

Location-aware Attention

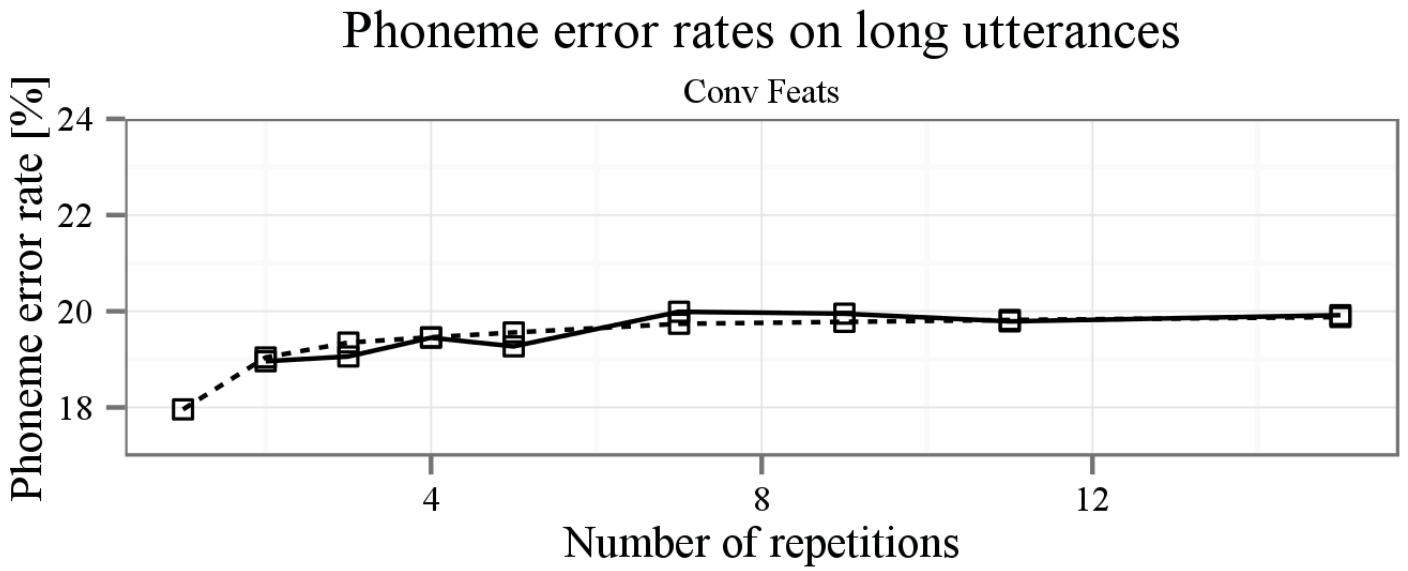


- We want to separate repetitions of the same sound
- Use the selection from the last step to make the new selection
- This enables the model to learn concepts like “later than last” or “close to last”.

Chorowski et al., „Attention-based models for speech recognition”, NIPS 2015

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

Chorowski et al., „Attention-based models for speech recognition”, NIPS 2015

Using Text-only Data

Text-only data is abundant. How to use it?

Pragmatic solution:

Add LM cost during beam search, find

$$Y = \arg\max_{\hat{Y}} \left(\log p_{\Theta}\left(\hat{Y}|X\right) + \alpha \log p_{LM}\left(\hat{Y}\right) - \gamma|\hat{Y}| \right)$$

On WSJ, WER reduction 18.6 -> 9.3

Problems: No clear probabilistic interpretation.

Bahdanau et al., „End-to-end attention-based large vocabulary speech recognition", ICASSP 2015

Alternatives to Cross-Entropy

We care about WER, not about perplexity.

Yet, we train with cross-entropy, the loss is:

$$-\sum_{i} \log p_{\Theta}(y_i | y_{i-1} \dots y_1, X)$$

There are two problems:

1. Teacher-forcing
2. The model does not learn from its mistakes

Better training criteria

For an input X, let Y^* be the best output.

Define a loss $l(Y, Y^*)$ measuring how bad is answering Y instead of Y^* (e.g. WER or BLEU).

We want to incorporate our knowledge of l into the training criterion.

Idea #1: Bahdanau et al, "Task Loss Estimation for Sequence Prediction", http://arxiv.org/abs/1511.06456

Oversimplifying idea: teach the model to compute

$$f(X, Y) = l(Y^*, Y)$$

A practical algorithm can be formulated for WER reward.

Intuition: teach the network which extensions of a hypothesis will increase the WER (help beam search!)

Define loss for a partial string (unfinished recognition):

$$lp(Y^*, Y) = \min_Z WER(Y^*, YZ)$$

To train the model:

1. Let $\hat{Y} = y_1 \dots y_N$ be a decoding (ground truth or beam search result)
2. Feet $\hat{Y}$ to the network. After seeing character y_i teach the net to output:
   $\mathrm{f}(\mathrm{y_i}|\mathrm{y}_{i-1} \dots y_1, X) \approx lp(Y^*, y_1 \dots y_i) - lp(Y^*, y_1 \dots y_{i-1})$

IDEA #2: Norouzi et al, "Reward AugmentedMaximum Likelihood for Neural Structured Prediction", NIPS 2016

Idea: we want

$$-\tau\mathbb{H}(p_\Theta(Y|X)) + \mathbb{E}_{Y\sim p_\Theta(\mathrm{Y|X})} l(Y^*, Y) =$$
$$\tau D_{KL}(p_\Theta(Y|X) \parallel q(Y|Y^*, \tau)) + const$$

Where $q(Y|Y^*, \tau) = \frac{1}{Z}\exp\left(-\frac{l(Y^*,Y)}{\tau}\right)$

Instead we optimize:

$$D_{KL}\big(q(Y|Y^*, \tau) \parallel p_\Theta(Y|X)\big) =$$
$$\mathbb{E}_{Y\sim q(Y|Y^*,\tau)}[\log p_\Theta(Y|X)] + const$$

Gradient is easy! SGD friendly! Just use:

$$\mathbb{E}_{Y\sim q(Y|Y^*,\tau)}[\nabla \log p_\Theta(Y|X)]$$

On-line decoding

- Two issues:
  - Encoder uses BiRNNs, switching to plain LSTMs raises the WER by a few percentage points
  - At each step, the attention mechanism scans the whole input sequence

Windowing: Pragmatic on-line decoding

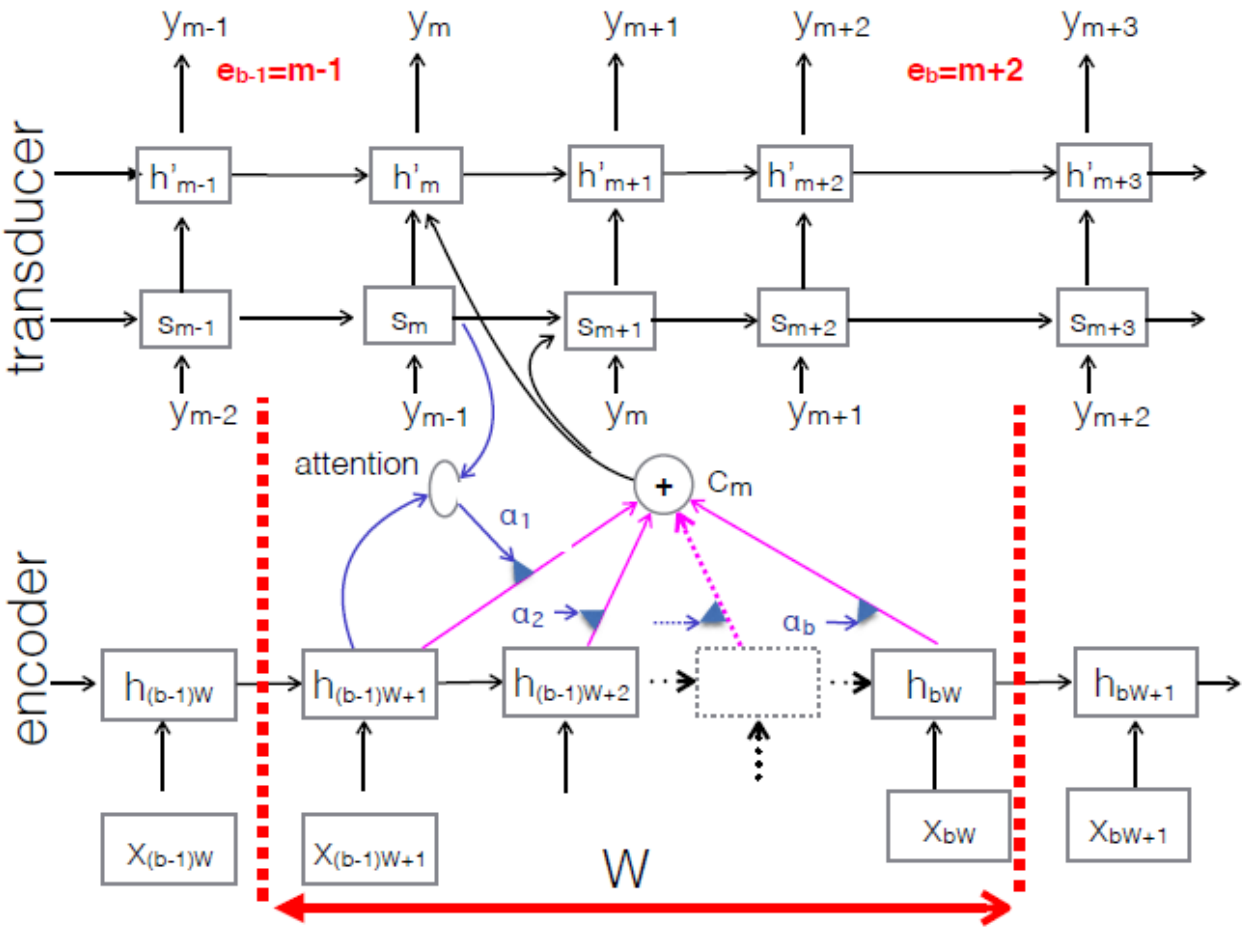
- Limit the attention mechanism to consider a small neighborhood of the mean/median location from the last step.
  - Let $p_t = \text{median}(\alpha)$ from previous step
  - Restrict new α to be 0 outside of $p_t \pm w$
- Proved to be the best trick in scaling to long utterances
- May require window application during training too

Bahdanau et al., „End-to-end attention-based large vocabulary speech recognition", ICASSP 2015

More principled on-line decoding

N. Jaitly et al, “A Neural Transducer”, NIPS 2016

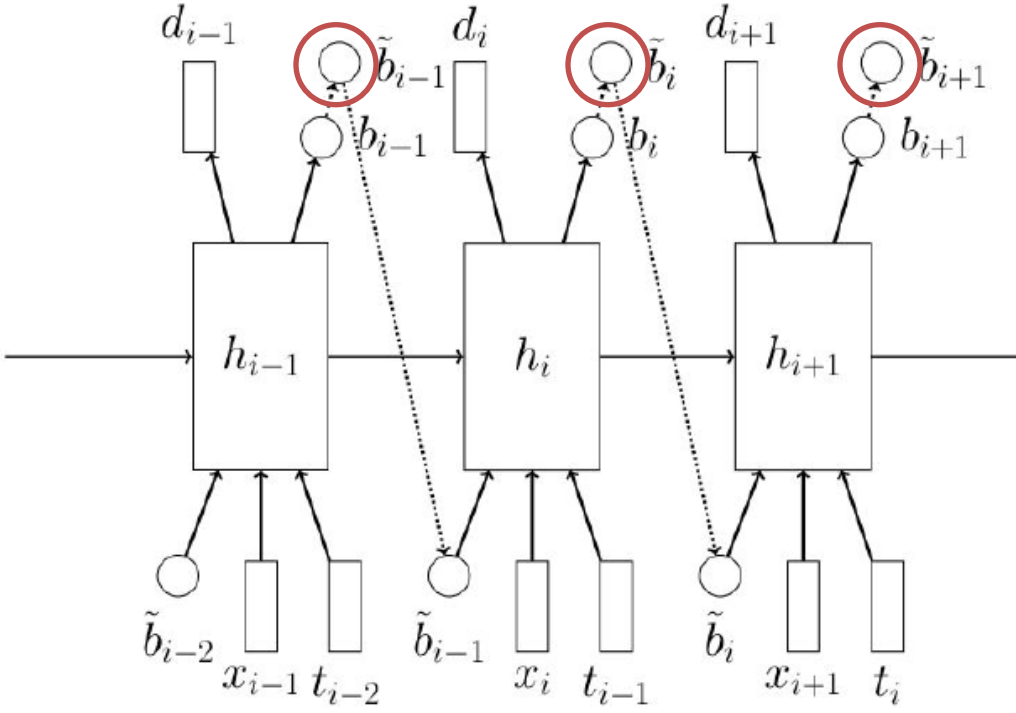
Core idea: Process the sequence in blocks



More-principled on-line decoding: signaling symbol emissions

Luo et al, “Learning Online Alignments with Continuous Rewards Policy Gradient”, arxiv.org/abs/1608.01281

- Decoder makes a step for each speech frame
- At every step, it chooses whether to emit a symbol
- Discrete decisions mandate RL training with Reinforce



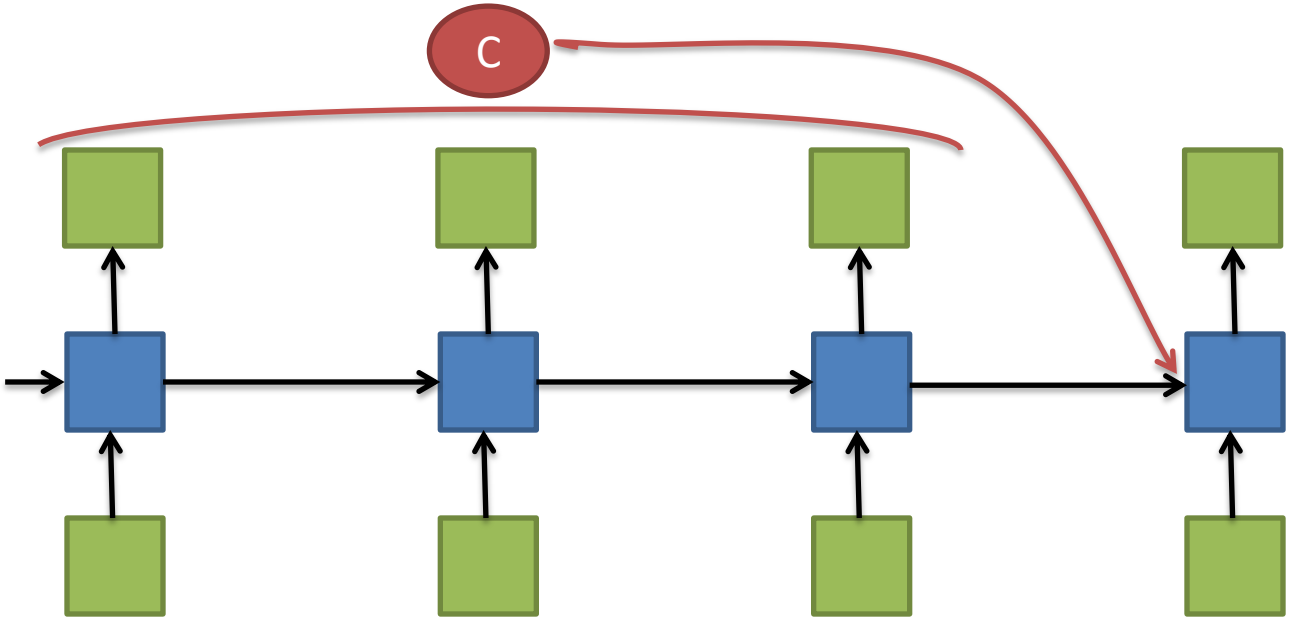
Attention for NLP

- Better language models
- Attention-based parsers

Building Better Language Models

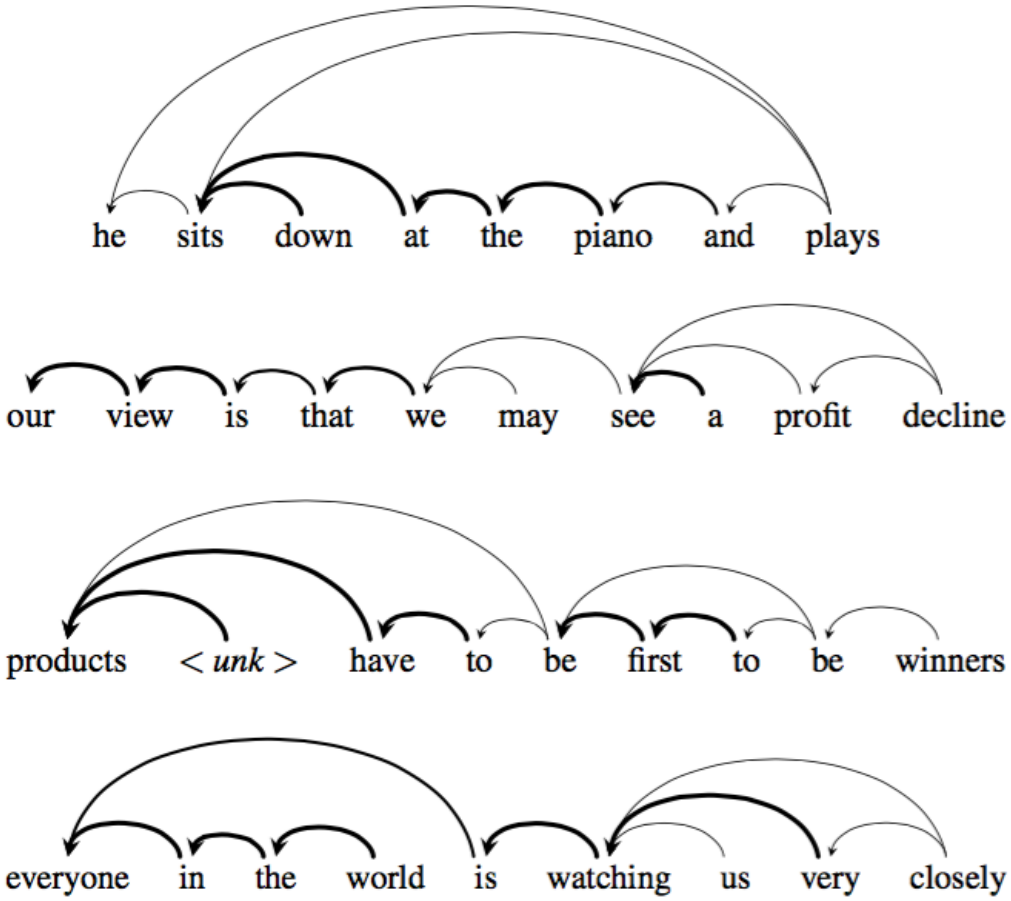
Big-picture idea:

- Take a recurrent language model
- At each step scan all previous inputs (or network states) using attention

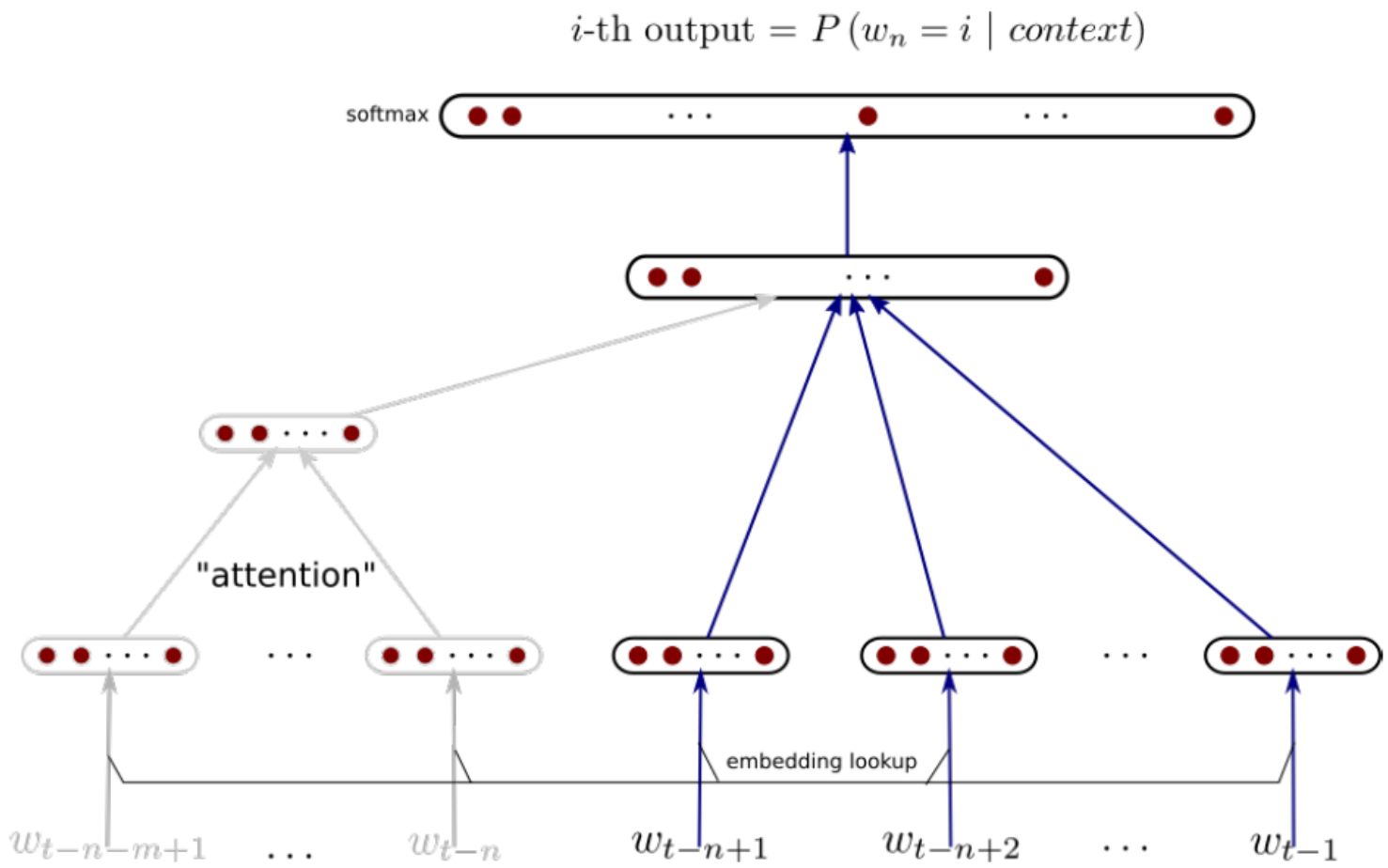


Attention in Language Model

J. Cheng et al, “Long Short-Term Memory-Networks for Machine Reading”, https://arxiv.org/abs/1601.06733



Autoregressive Model + Attention



m additional words are aggregated into a context using attention

n words are directly fed into the network

Autoregressive Model + Attention

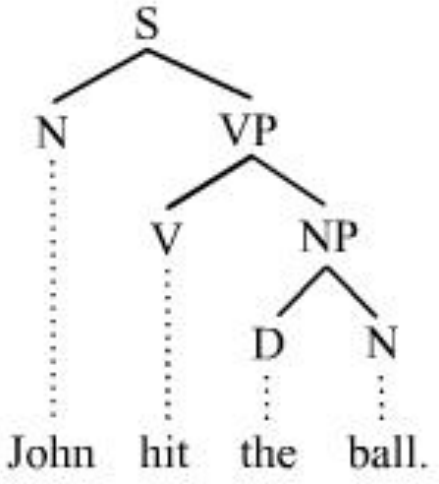
BRIAN ROSNER AN ASSISTANT DISTRICT ATTORNEY IN MANHATTAN
WHO EVENTUALLY PROSECUTED MESSRS. GRAMBLING AND LIBMAN

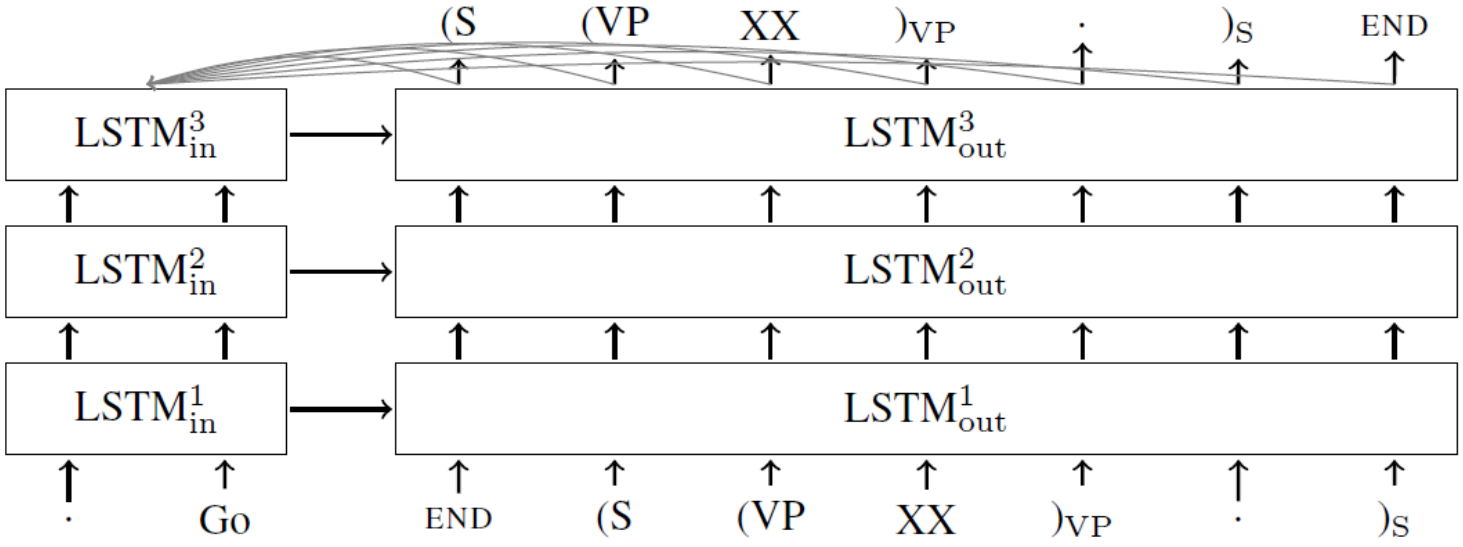
BRIAN ROSNER AN ASSISTANT DISTRICT
ROSNER AN ASSISTANT DISTRICT ATTORNEY
AN ASSISTANT DISTRICT ATTORNEY IN
ASSISTANT DISTRICT ATTORNEY IN MANHATTAN
DISTRICT ATTORNEY IN MANHATTAN WHO
ATTORNEY IN MANHATTAN WHO EVENTUALLY
IN MANHATTAN WHO EVENTUALLY PROSECUTED
MANHATTAN WHO EVENTUALLY PROSECUTED MESSRS.
WHO EVENTUALLY PROSECUTED MESSRS. GRAMBLING
EVENTUALLY PROSECUTED MESSRS. GRAMBLING AND
PROSECUTED MESSRS. GRAMBLING AND LIBMAN

How to parse sentences?

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

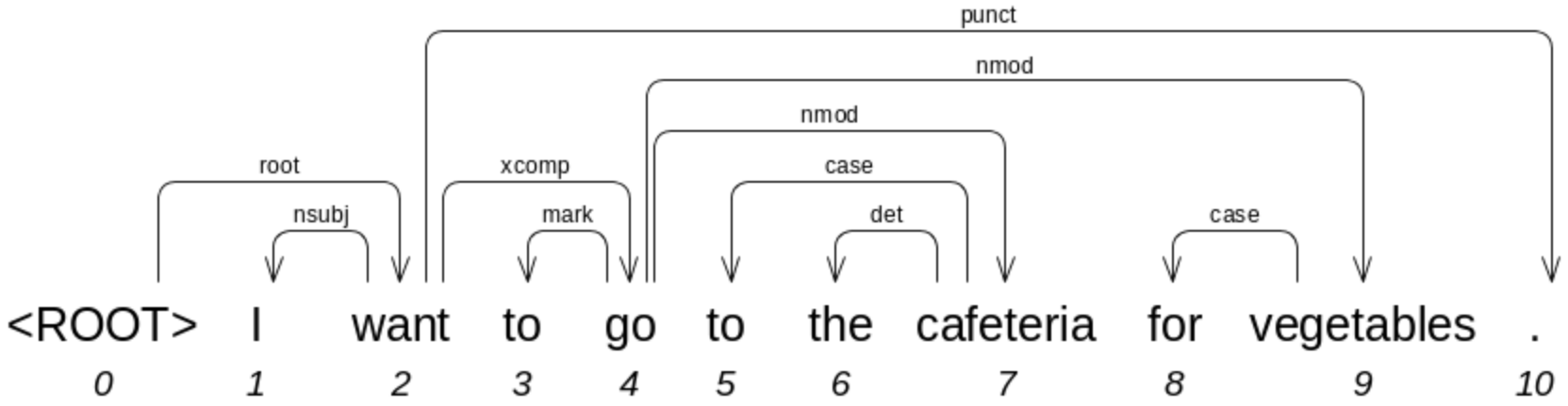
- Input: sentence
  „Go .”
- Output: linearized parse tree:
  „(S (VP XX)VP .)S END”





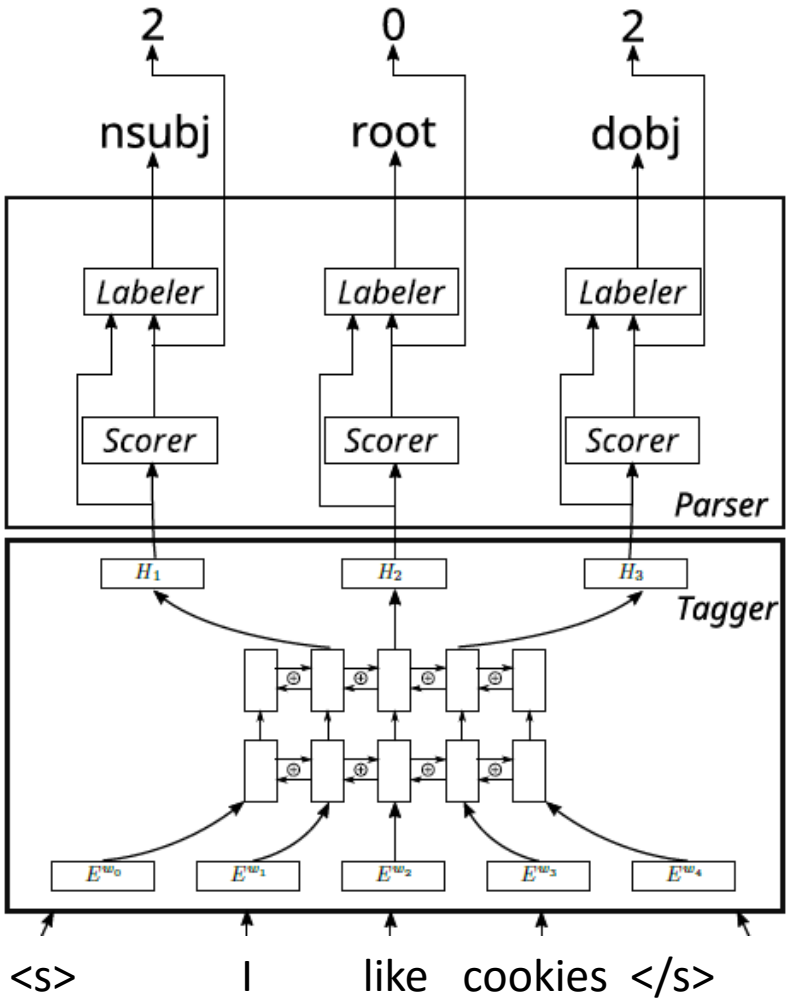
O. Vinyals et al, “Grammar as a Foreign Language”, NIPS 2015

Dependency parsing



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

Dependency parsing

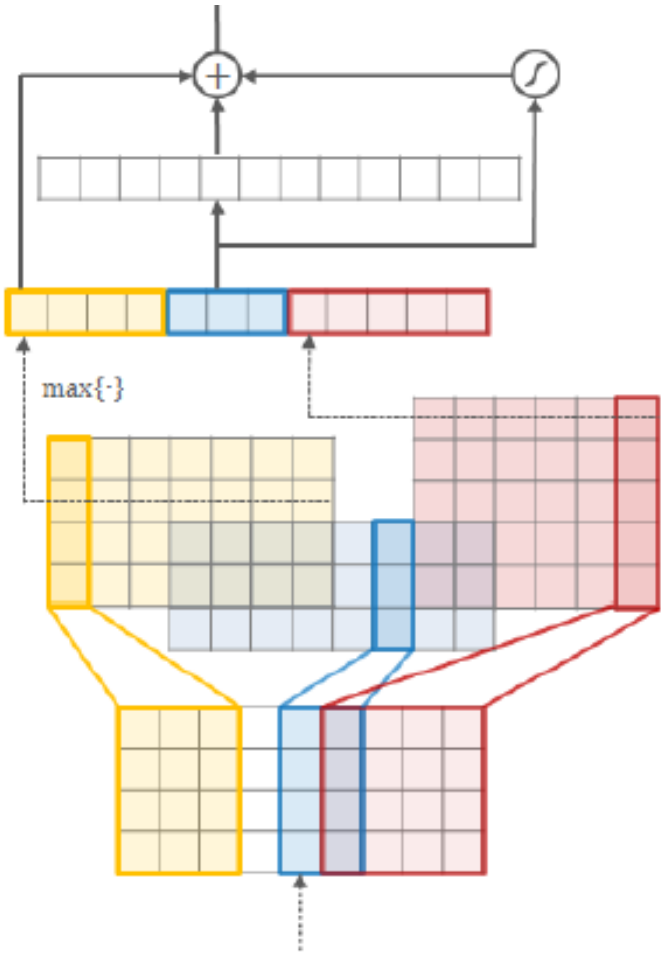


For each word w

Two operations:

1. Find head h (use attention mechanism)
2. Use (w, h) to predict dependency type

From characters to word embeddings



„incomprehensible”

Highway layers – very nonlinear transformation of data

Glue best word representations together

Convolutional filters of varying lengths. Can react to pre-, in-, and post-fixes of words

Character embeddings concatenated into a matrix

Y. Kim, Y. Jernite, D. Sontag, and A. M. Rush, “Character-Aware Neural Language Models,” *arXiv:1508.06615 [cs, stat]*, Aug. 2015.

Parser results

Results on common dependencies treebanks

	Czech			English			Polish		
	LA	UAS	LAS	LA	UAS	LAS	LA	UAS	LAS
Gold POS tags									
MaltParser	91.6	86	83	92.0	87.0	84.0	92.0	89.1	85.8
(Straka et al., 2015)	-	87.7	84.7	-	88.2	84.8	-	89.8	85.5
(Tiedemann, 2015)	-	-	85.7	-	-	**85.7**	-	-	-
NN (this work)	**93.8**	**91.7**	**88**	**92.3**	**88.6**	85.1	**93.9**	**93.4**	**89.3**
Predicted POS tags or no POS tags									
(Tiedemann, 2015)	-	-	81.4	-	-	82.7	-	-	-
NN words	82.4	82.4	72.1	85	81.9	74.7	73.8	74.6	61.6
NN chars, soft att.	92.1	**90.1**	85.7	90	86.5	82.1	88.7	89.1	82.5
NN chars, tags, soft att.	89.5	89.6	82.8	89.2	86.2	81.3	89.3	90.4	83.9
NN chars, tags, hard att.	**92.6**	**90.1**	**86.7**	**90.4**	**87.6**	**83.6**	**90.9**	**91.3**	**86**

Note: MaltParser results on Czech are sub-optimal because due to lack of computational resources we had to use a small dataset for parser optimization.

http://arxiv.org/abs/1609.03441

Thank you

We have code:
https://github.com/rizar/attention-lvcsr

Questions?

References & Further Reading

- Good review presentation: T. Sainath “Towards End-to-End speech recognition using deep networks” https://sites.google.com/site/tsainath/
- D. Amodei, el 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
- 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
- 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
- Norouzi et al, “Reward AugmentedMaximum 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
- 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., Sutskver, I., Hinton, G., Grammar as a Foreign Language, NIPS 2015
- Vinyals, O., Fortunato, M., Jaitly, N., Pointer Networks, NIPS 2015