

Natural Language Processing with Deep Learning

CS224N/Ling284



Lecture 18:
Tackling the Limits of Deep Learning for NLP

Christopher Manning and Richard Socher

Update on PA4 and Final Project

- Sorry and Thanks!
- Being on the cutting edge (at scale) is hard
- We are giving a short (33 hour!) extension for Ass4/
Final Project: New deadline: 9am Sunday, 3/19
- Hard deadline (with use of late days/penalties): 9am
Wednesday, 3/22
- Poster session! → 5% of final grade.
- Final Ass4/FP: 27%

Update on PA4 and Final Project

- More helper **pseudo**-code with a simple model to get you off the ground.
- Another simple baseline you can try (>50% F1)

question -> LSTM -> q

input -> LSTM -> x_i at each word

`classify_with_neural_net(q, xi)` to predict start

`classify_with_neural_net(q, xi)` to predict end

Poster Session

- Everyone expected to attend (or video).
- Check out other student's posters.
- Lunch on us.
- Jobs and funding :)

The Limits of Single Task Learning

- Great performance improvements
- Projects start from random
- Single unsupervised task can't fix it
- How to express different tasks in the same framework, e.g.
 - sequence tagging
 - sentence-level classification
 - seq2seq?

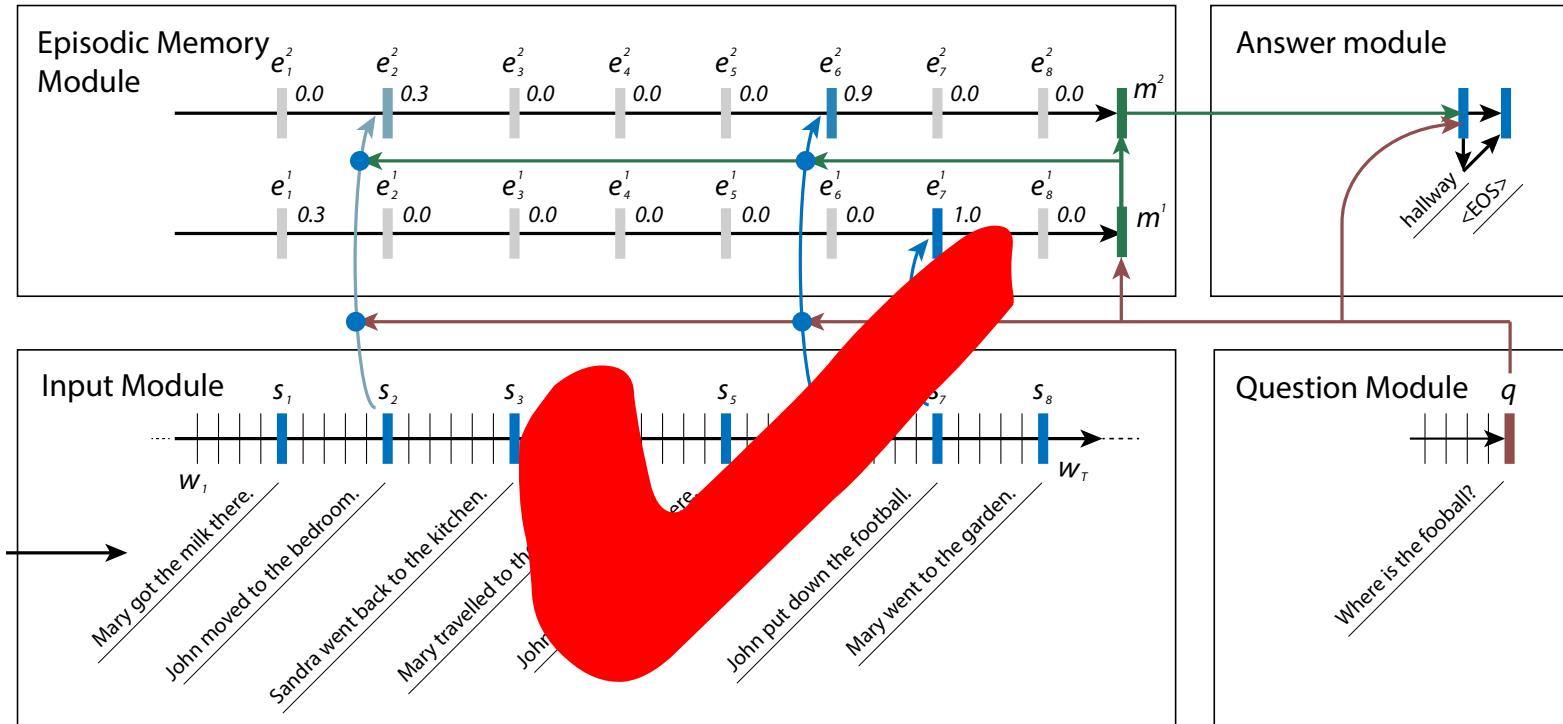


First of Several Major Obstacles

- For NLP no single model **architecture** with consistent state of the art results across tasks

Task	State of the art model
Question answering (babI)	Strongly Supervised MemNN (Weston et al 2015)
Sentiment Analysis (SST)	Tree-LSTMs (Tai et al. 2015)
Part of speech tagging (PTB-WSJ)	Bi-directional LSTM-CRF (Huang et al. 2015)

Tackling Obstacle 1: Dynamic Memory Network



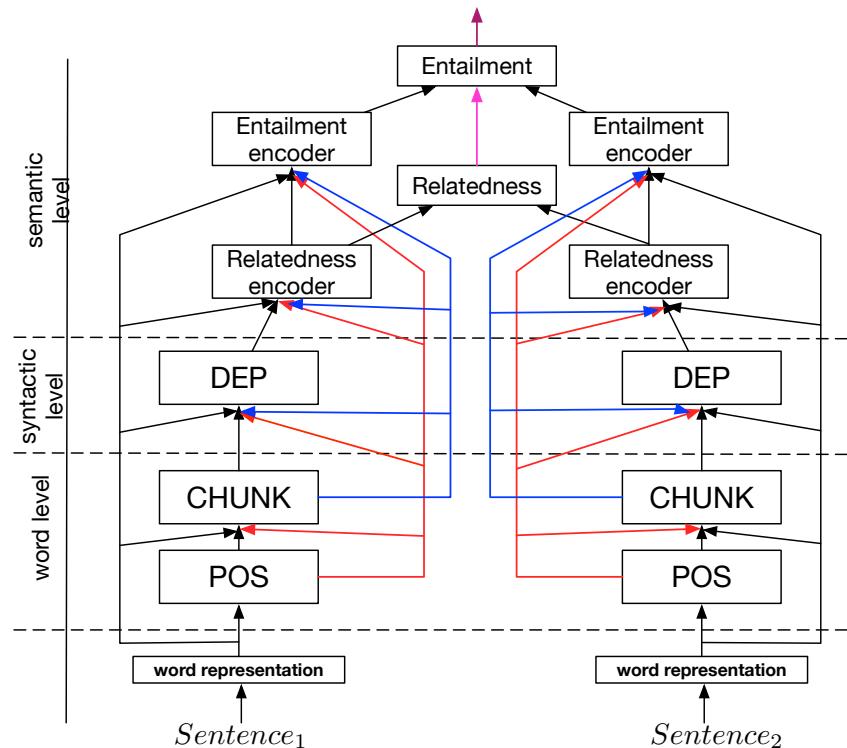
Obstacle 2: Joint Many-task Learning

- Fully joint multitask learning* is hard:
 - Usually restricted to lower layers
 - Usually helps only if tasks are related
 - Often hurts performance if tasks are not related

* meaning: same decoder/classifier
and not only transfer learning with source
target task pairs

Tackling Joint Training

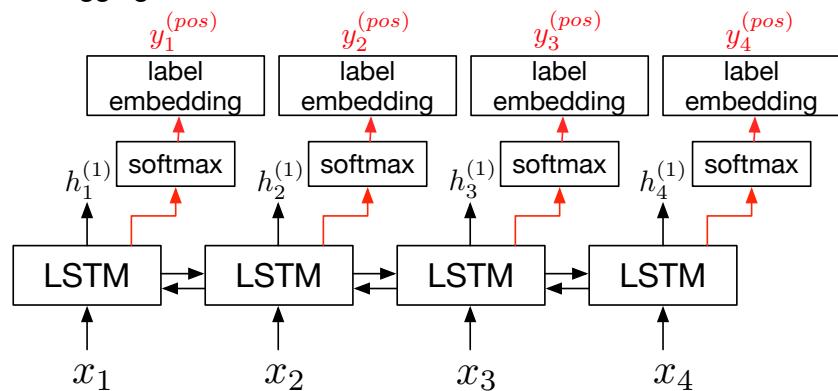
- A Joint Many-Task Model:
Growing a Neural Network for Multiple NLP Tasks
Kazuma Hashimoto,
Caiming Xiong,
Yoshimasa Tsuruoka &
Richard Socher
- Final Model →



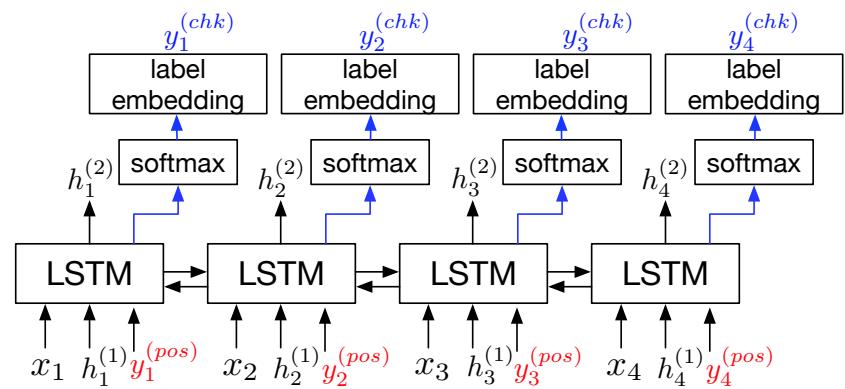
Model Details

- Include character n-grams and short-circuits
- State of the art purely feedforward parser

POS Tagging:



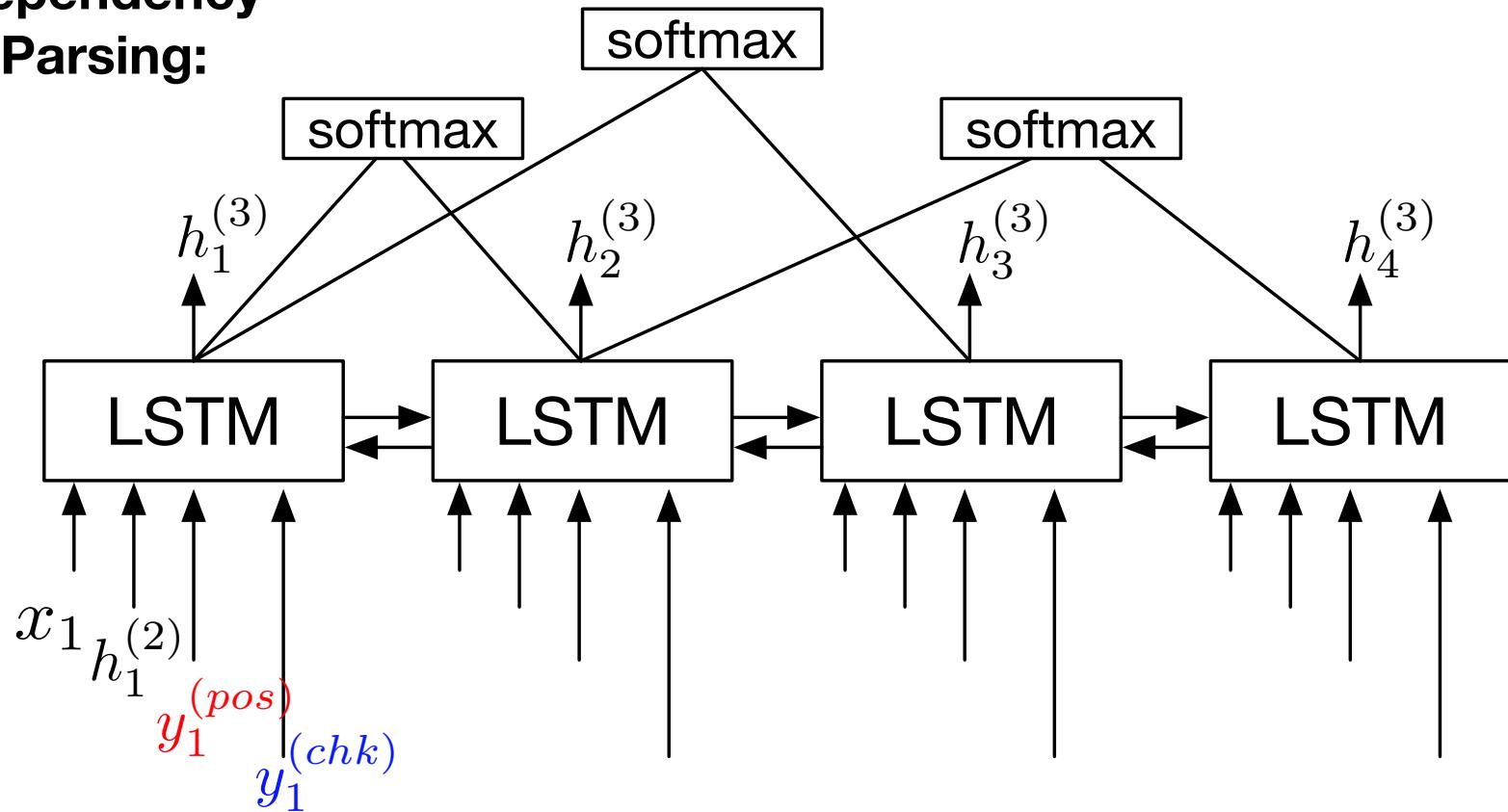
Chunking:



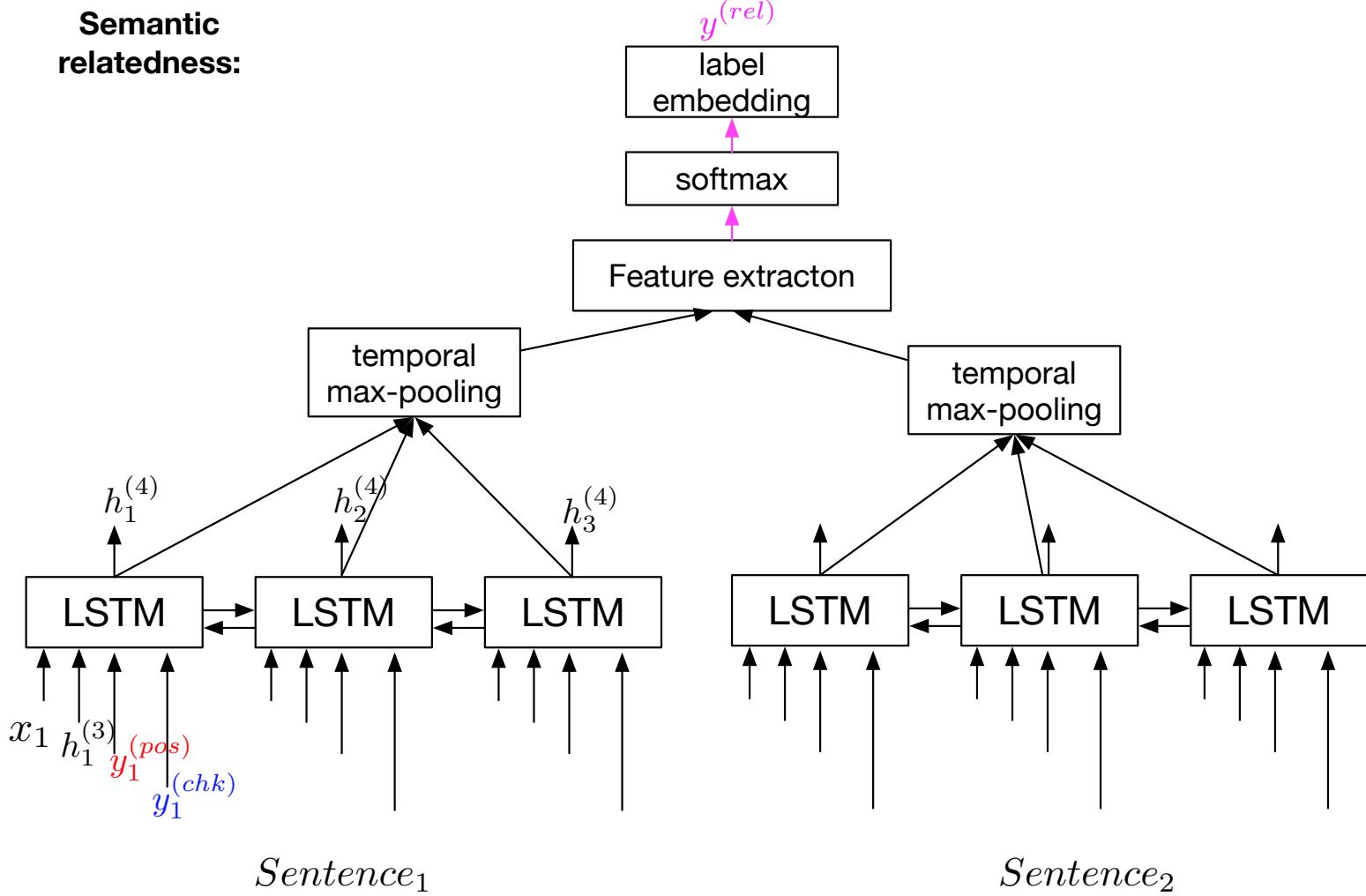
$$y_t^{(pos)} = \sum_{j=1}^C p(y_t^{(1)} = j | h_t^{(1)}) \ell(j),$$

Dependency Parsing

Dependency
Parsing:



Multi Sentence Tasks: Semantic Relatedness



Training Details: Regularized Idea

Chunking training

$$-\sum_s \sum_t \log p(y_t^{(2)} = \alpha | h_t^{(2)}) + \lambda \|W_{\text{chunk}}\|^2 + \delta \|\theta_{\text{POS}} - \theta'_{\text{POS}}\|^2,$$

Entailment training

$$-\sum_{(s,s')} \log p(y_{(s,s')}^{(5)} = \alpha | h_s^{(5)}, h_{s'}^{(5)}) + \lambda \|W_{\text{ent}}\|^2 + \delta \|\theta_{\text{rel}} - \theta'_{\text{rel}}\|^2,$$

Joint Training Helps Here!

		Single	JMT _{all}	JMT _{AB}	JMT _{ABC}	JMT _{DE}
A	POS	97.45	97.55	97.52	97.54	n/a
B	Chunking	95.02	(97.12)	95.77	(97.28)	n/a
C	Dependency UAS	93.35	94.67	n/a	94.71	n/a
	Dependency LAS	91.42	92.90	n/a	92.92	n/a
D	Relatedness	0.247	0.233	n/a	n/a	0.238
E	Entailment	81.8	86.2	n/a	n/a	86.8

New State of the Art on 4 of 5 Tasks

Method	Acc.
JMT _{all}	97.55
Ling et al. (2015)	97.78
Kumar et al. (2016)	97.56
Ma & Hovy (2016)	97.55
Søgaard (2011)	97.50
Collobert et al. (2011)	97.29
Tsuruoka et al. (2011)	97.28
Toutanova et al. (2003)	97.27

Table 2: POS tagging results.

Method	F1
JMT _{AB}	95.77
Søgaard & Goldberg (2016)	95.56
Suzuki & Isozaki (2008)	95.15
Collobert et al. (2011)	94.32
Kudo & Matsumoto (2001)	93.91
Tsuruoka et al. (2011)	93.81

Table 3: Chunking results.

Method	UAS	LAS
JMT _{all}	94.67	92.90
Single	93.35	91.42
Andor et al. (2016)	94.61	92.79
Alberti et al. (2015)	94.23	92.36
Weiss et al. (2015)	93.99	92.05
Dyer et al. (2015)	93.10	90.90
Bohnet (2010)	92.88	90.71

Table 4: Dependency results.

Method	MSE
JMT _{all}	0.233
JMT _{DE}	0.238
Zhou et al. (2016)	0.243
Tai et al. (2015)	0.253

Table 5: Semantic relatedness results.

Method	Acc.
JMT _{all}	86.2
JMT _{DE}	86.8
Yin et al. (2016)	86.2
Lai & Hockenmaier (2014)	84.6

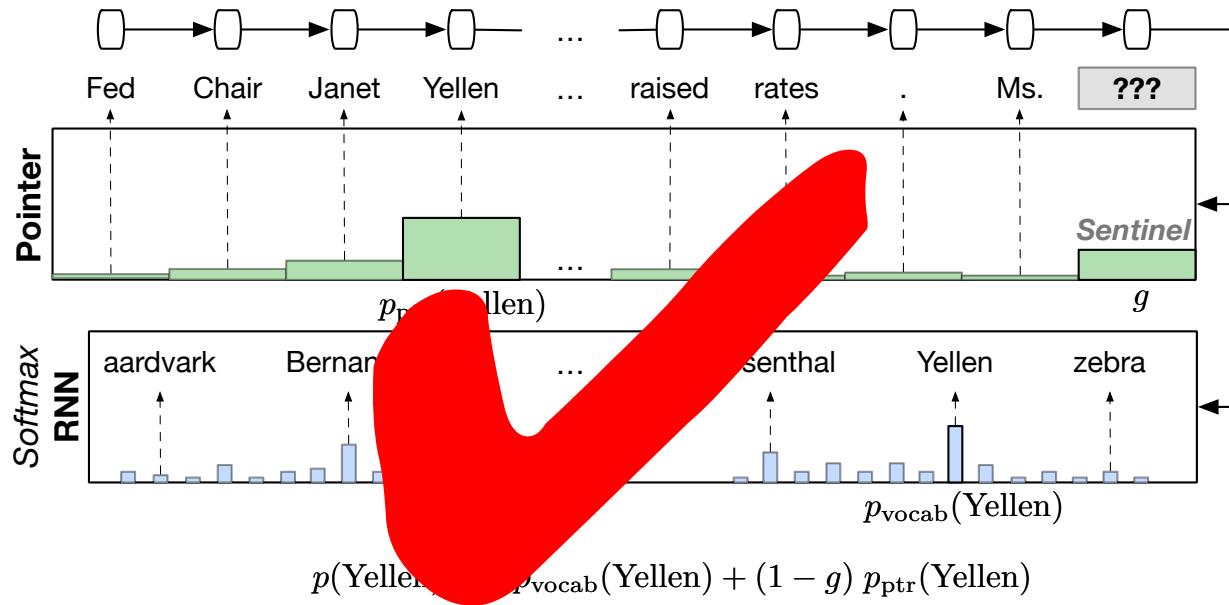
Table 6: Textual entailment results.

Obstacle 3: No Zero Shot Word Predictions

- Answers can only be predicted if they were seen during training and part of the softmax
- But it's natural to learn new words in an active conversation and systems should be able to pick them up

Tackling Obstacle by Predicting Unseen Words

- Idea: Mixture Model of softmax and pointers:



- Pointer Sentinel Mixture Models by Stephen Merity, Caiming Xiong, James Bradbury, Richard Socher

Neural Turing Machines

Presented by Nish Khandwala

Papers covered:

- "Neural turing machines."
Graves, Alex, Greg Wayne, and Ivo Danihelka.
arXiv:1410.5401 (2014).
- "Hybrid computing using a neural network
with dynamic external memory."
Graves, Alex, et al. *Nature* 538.7626 (2016):
471-476.

The Problem

The neural networks (NN) we have seen in class so far excel at pattern recognition and reactive decision making but are unable to *deliberate or reason using knowledge.*

Easy for NN



Playing
Breakout

Hard for NN

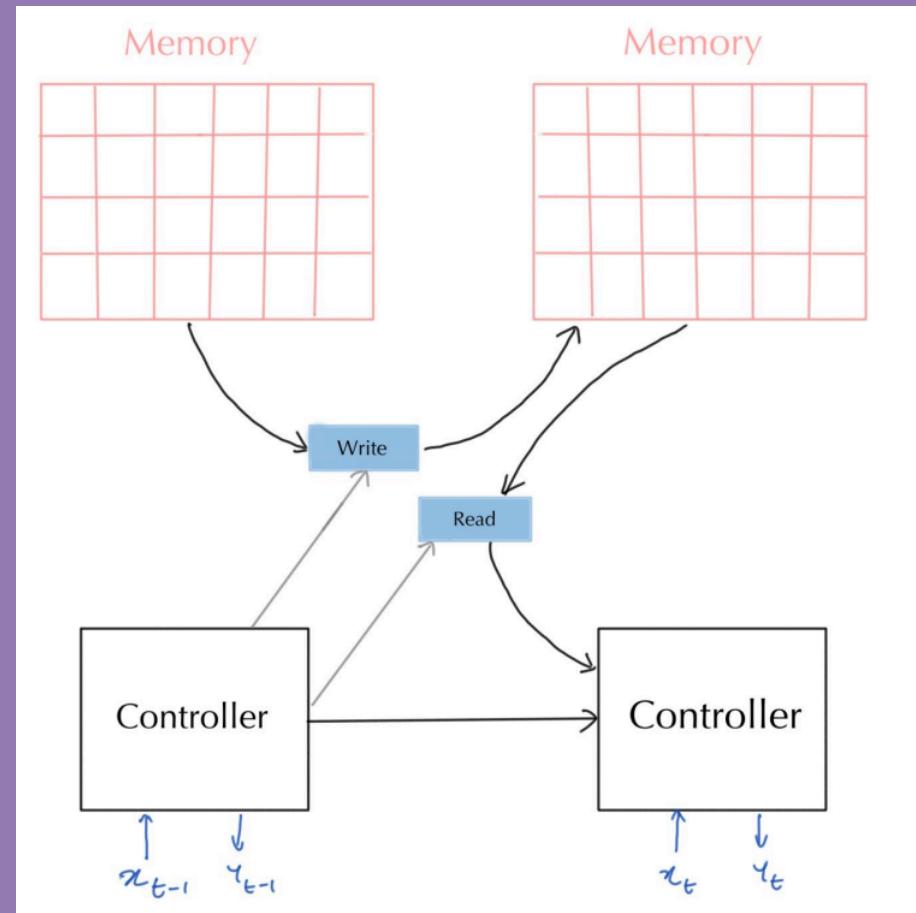


Finding Shortest
Path

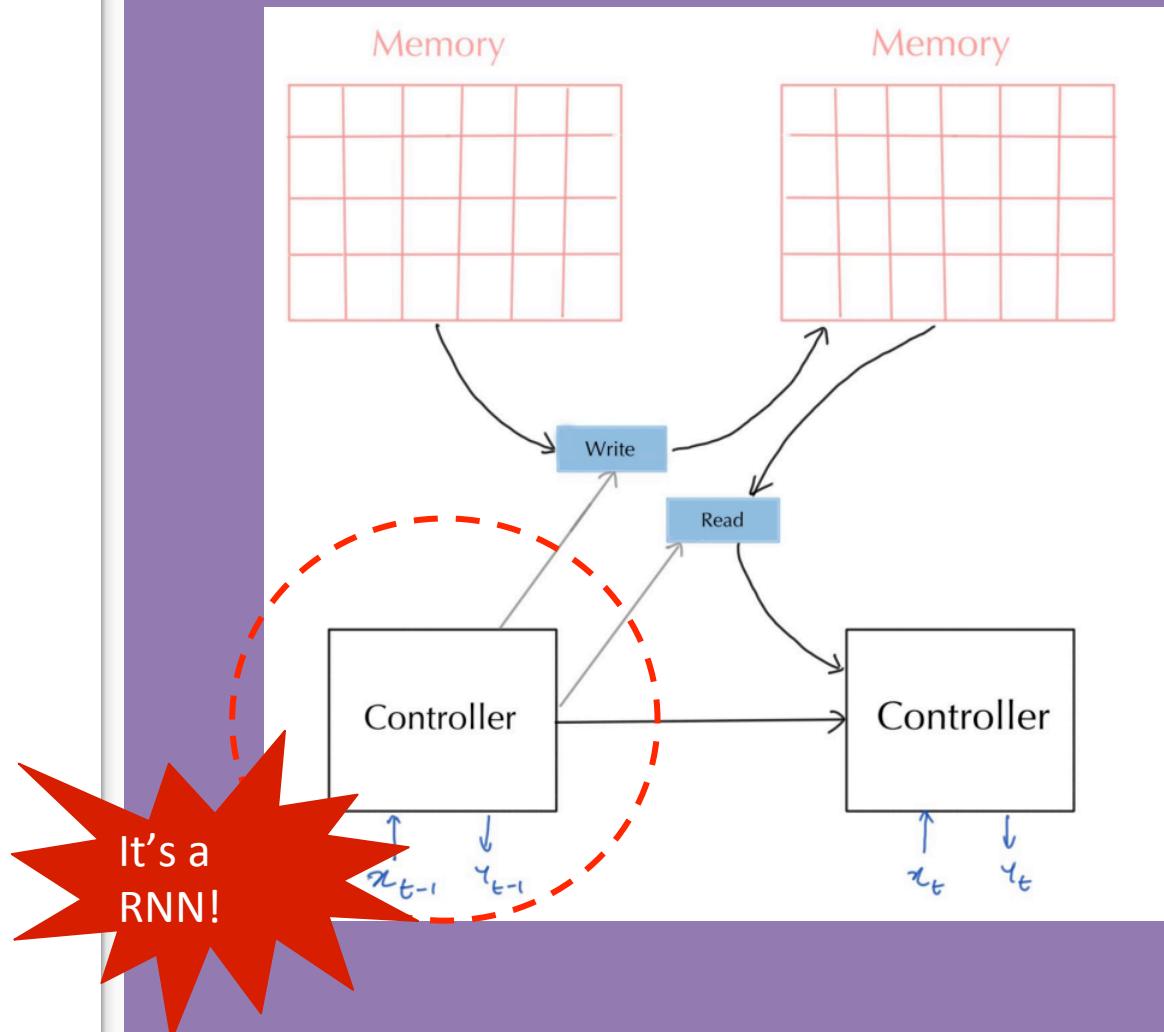
The Solution? Memory...

Wait, I thought LSTMs had memory cells...?
Yes, but we need an *external* memory bank. If a
LSTM's memory cell is a cache, what we need
is RAM.

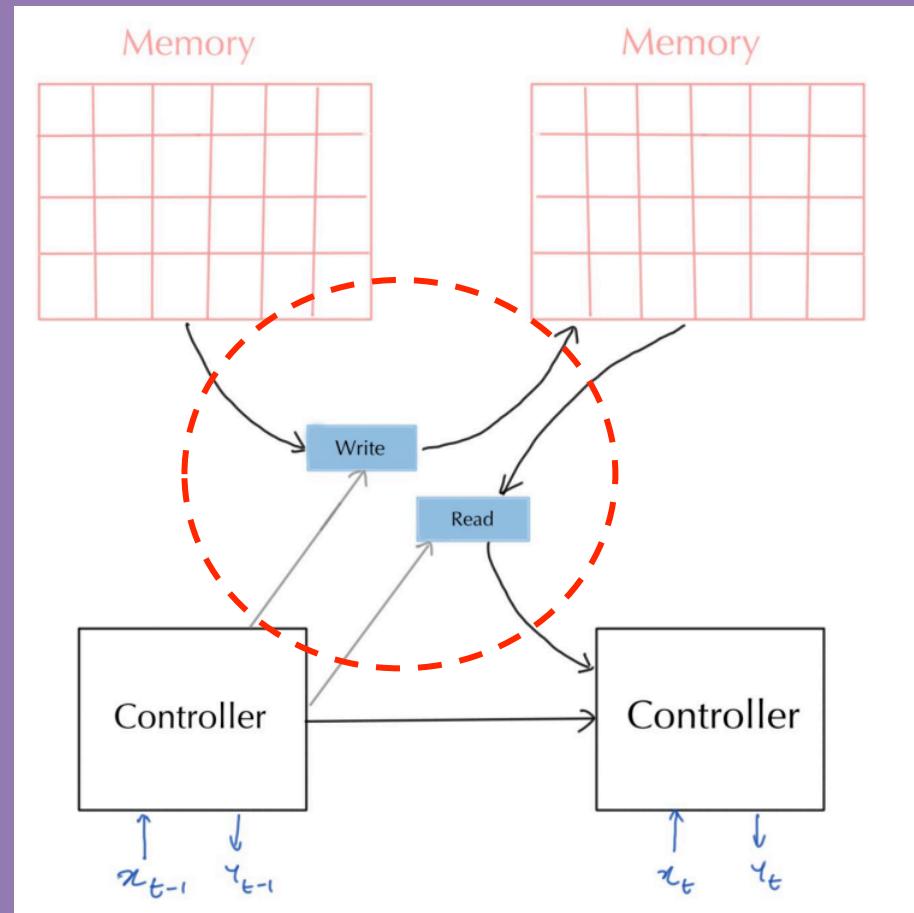
Neural Turing Machines



Neural Turing Machines



Neural Turing Machines



How does reading and writing work?

Reading from and writing to memory should be differentiable with respect to the location we need to operate on.

But this is tricky! Why?

How does reading and writing work?

Reading from and writing to memory should be differentiable with respect to the location we need to operate on.

But this is tricky! Why? Memory is fundamentally discrete. :(

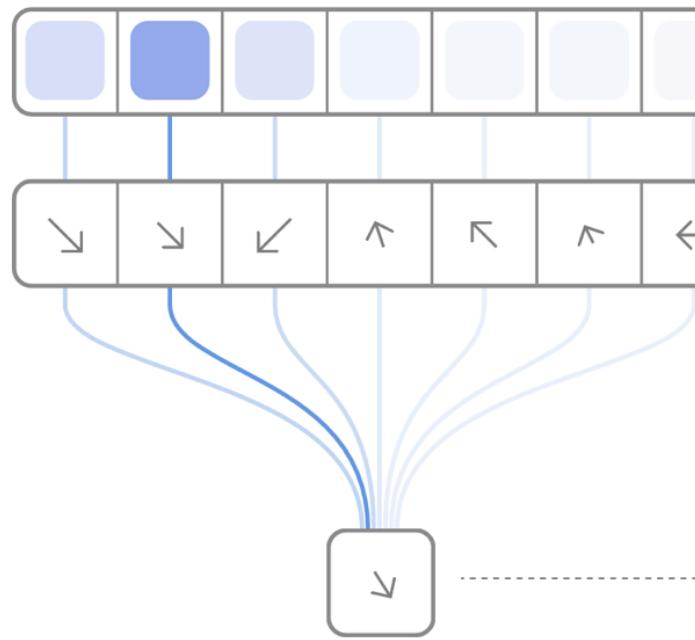
How does reading and writing work?

Reading from and writing to memory should be differentiable with respect to the location we need to operate on.

But this is tricky! Why? **Memory is fundamentally discrete. :(**

Solution? Read and write everywhere, just to different extents.. (think attention)

Reading from Memory



attention

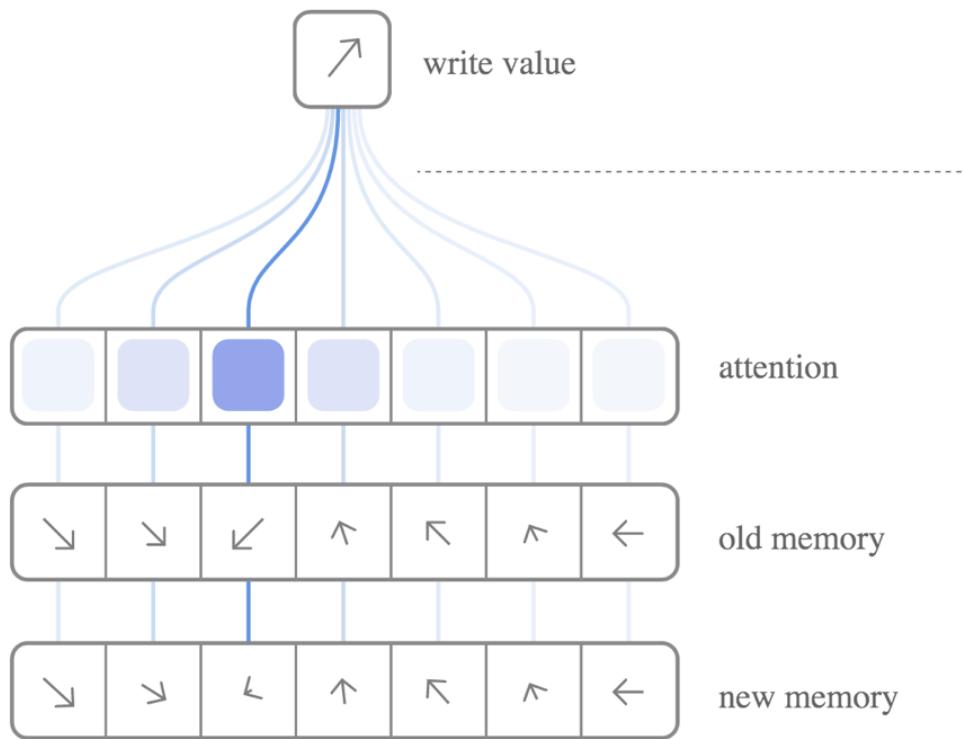
memory

The RNN gives an attention distribution which describe how we spread out the amount we care about different memory positions

The read result is a weighted sum.

$$r \leftarrow \sum_i a_i M_i$$

Writing to Memory

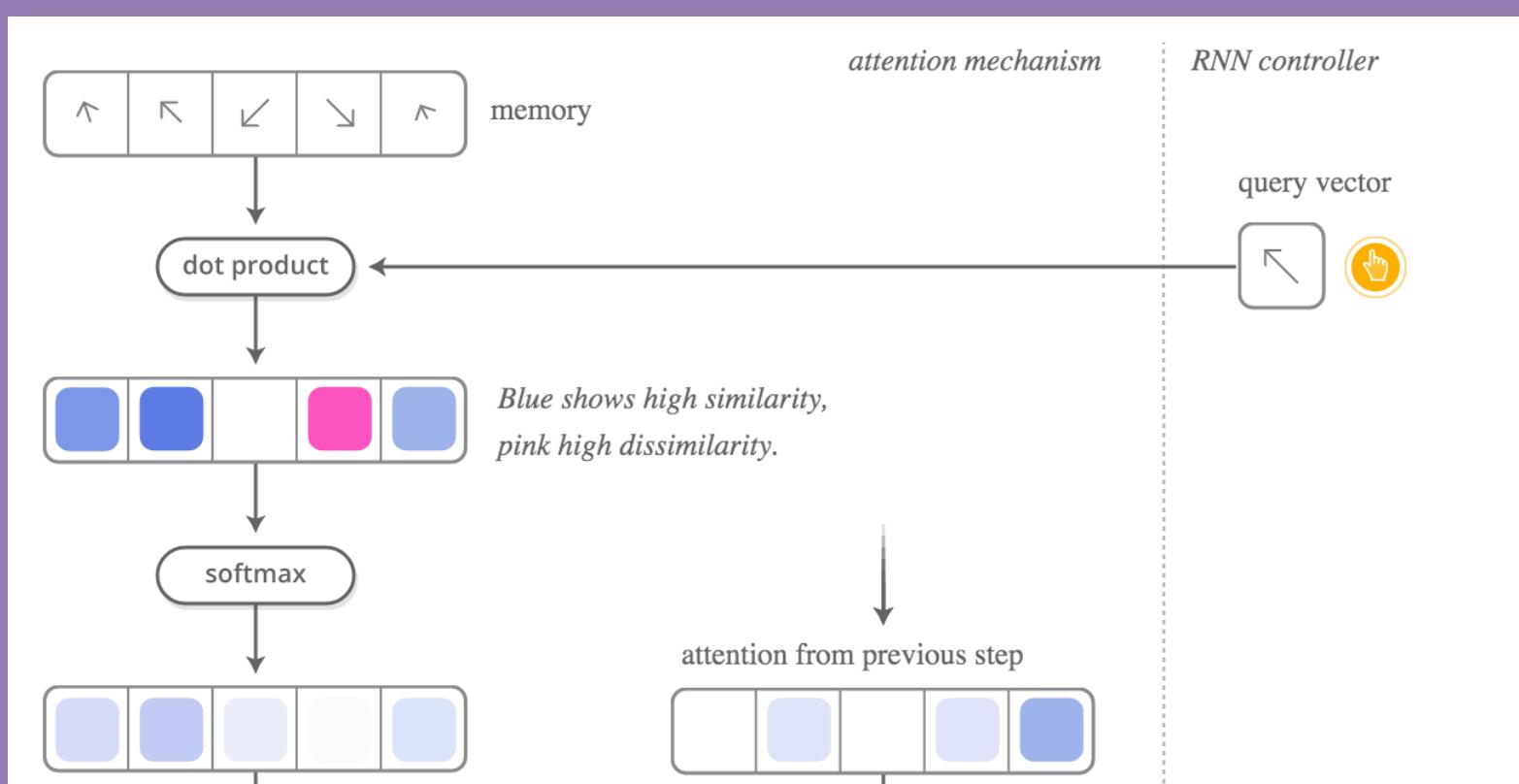


Instead of writing to one location, we write everywhere, just do different extents.

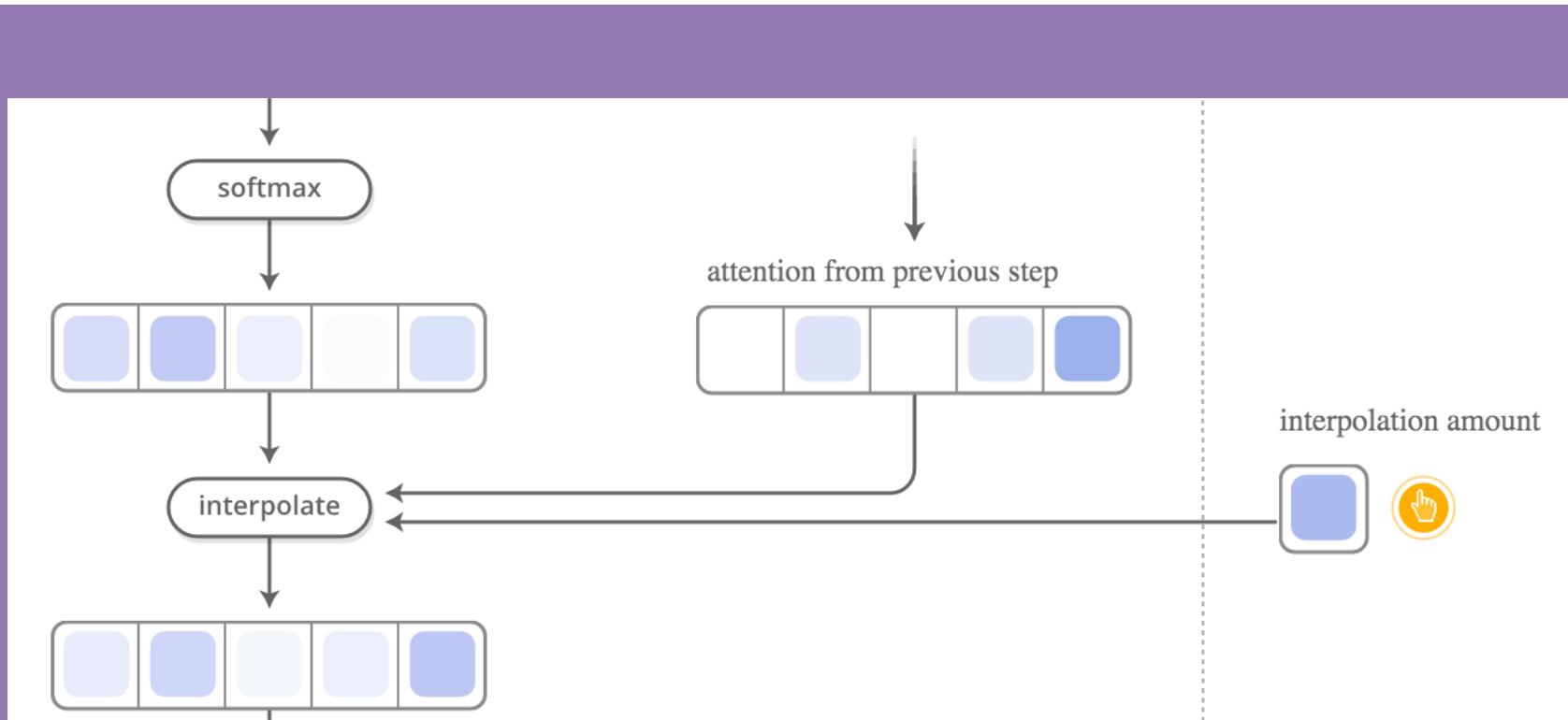
The RNN gives an attention distribution, describing how much we should change each memory position towards the write value.

$$M_i \leftarrow a_i w + (1-a_i) M_i$$

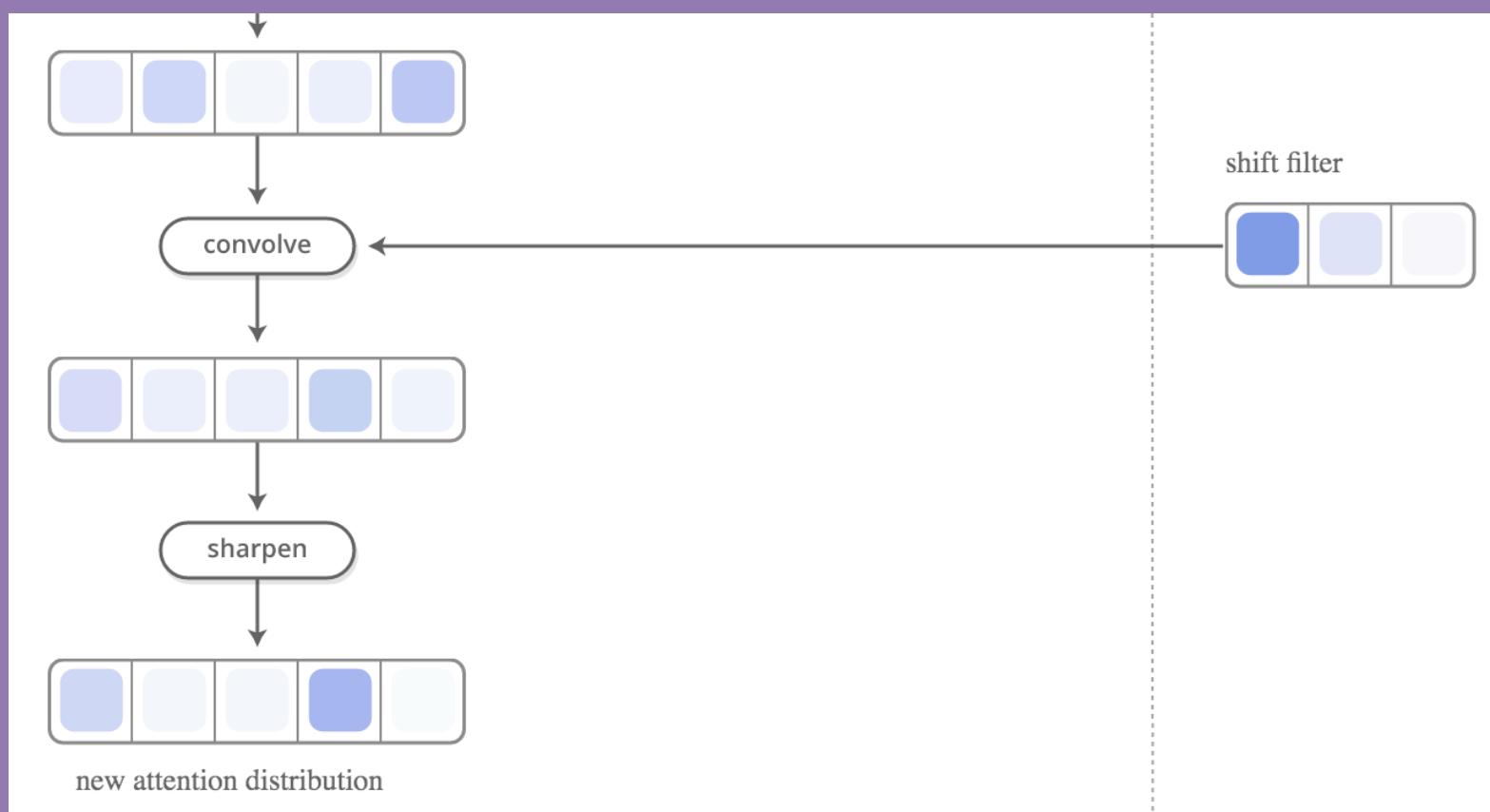
Updating Attention - 1/3



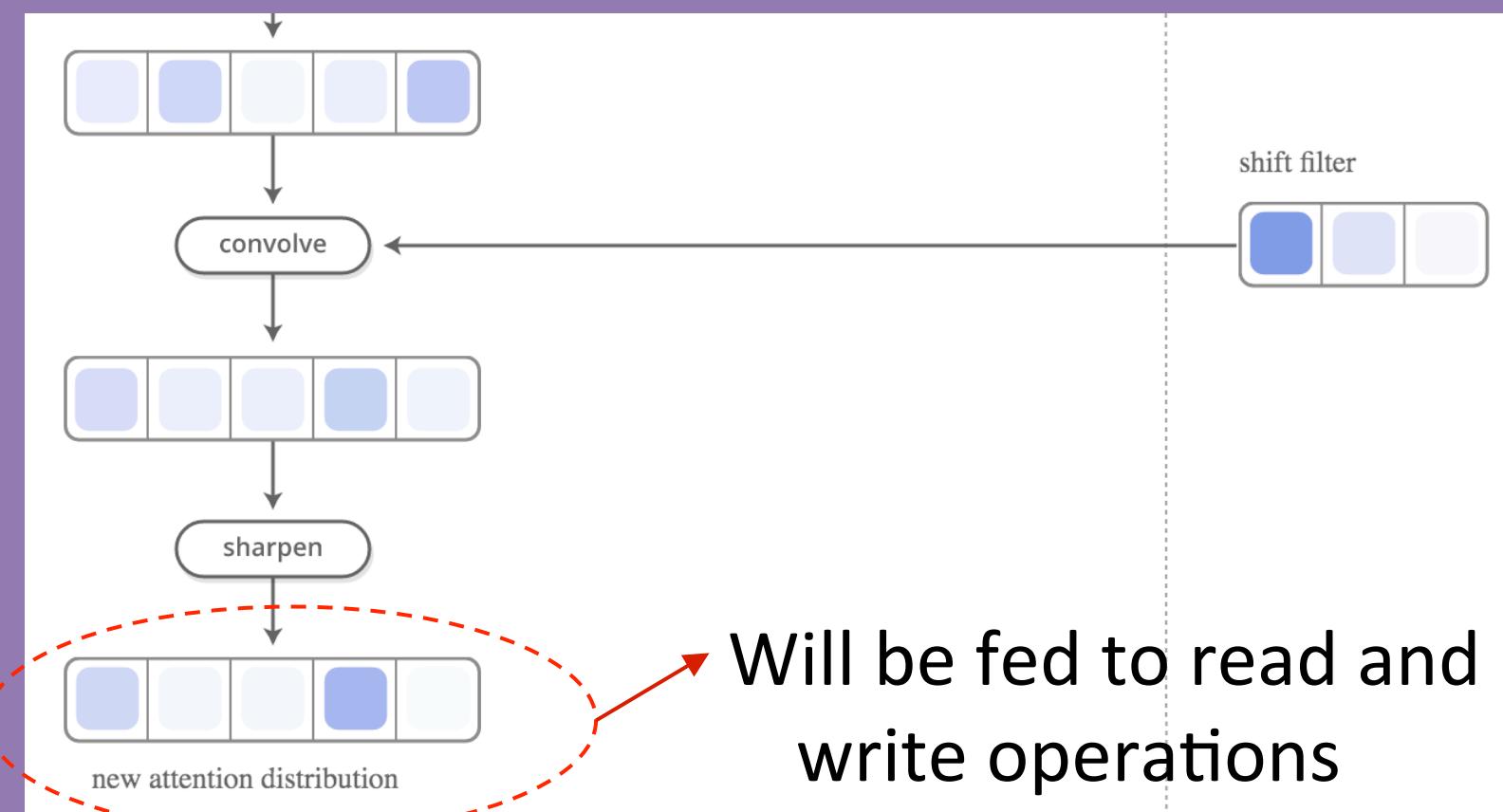
Updating Attention - 2/3



Updating Attention - 3/3



Updating Attention - 3/3



Will be fed to read and
write operations

Results*



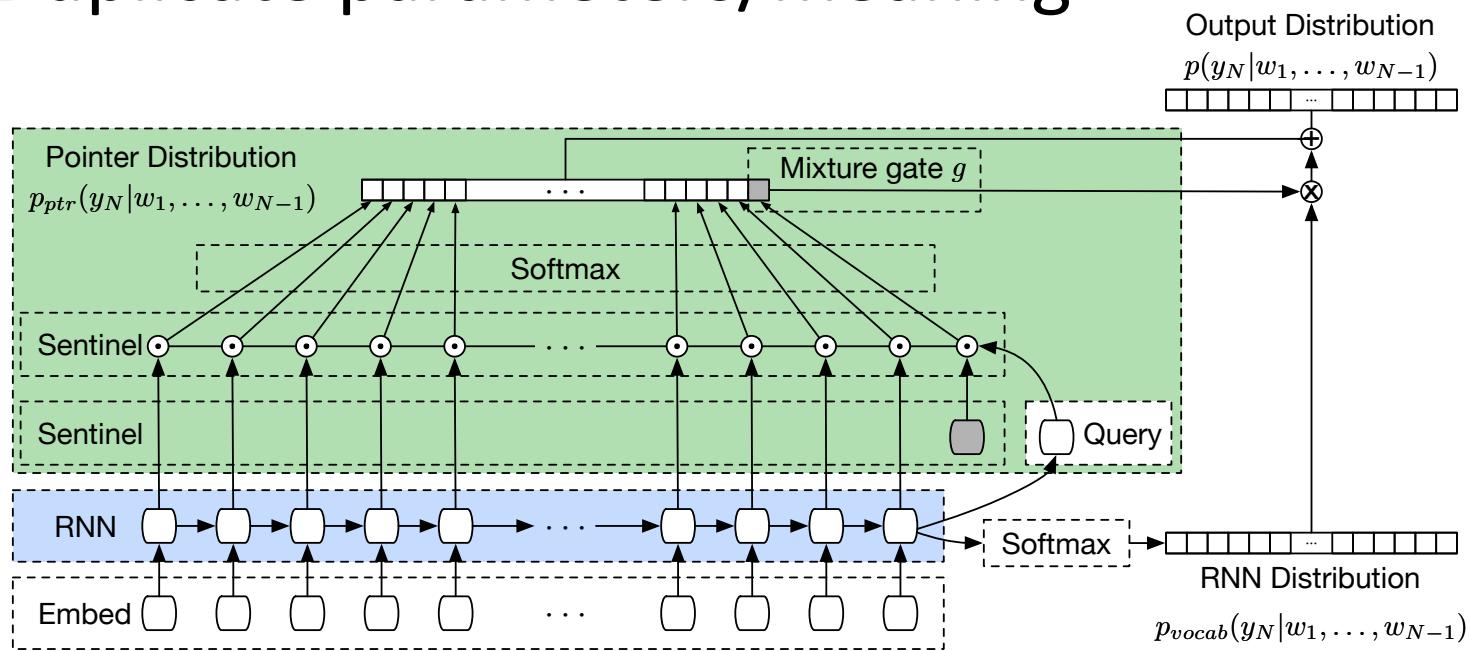
* using a differentiable neural computer (follow up architecture to NTMs)

Acknowledgements

- Olah & Carter, "Attention and Augmented Recurrent Neural Networks", Distill, 2016.
- <https://deepmind.com/blog/differentiable-neural-computers/>

Obstacle 4: Duplicate Word Representations

- Different encodings for encoder (Word2Vec and GloVe word vectors) and decoder (softmax classification weights for words)
- Duplicate parameters/meaning



Tackling Obstacle by Tying Word Vectors

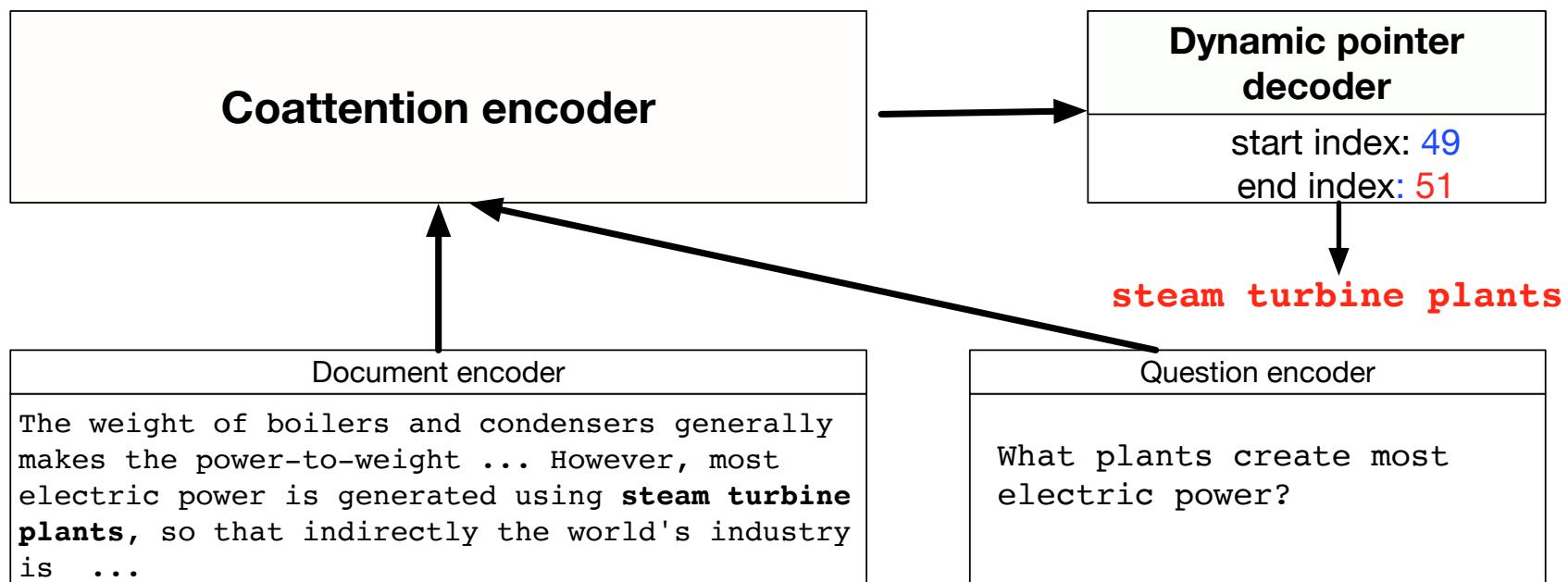
- Simple but theoretically motivated idea: tie word vectors and train single weights jointly
- Tying Word Vectors and Word Classifiers: A Loss Framework for Language Modeling, Hakan Inan, Khashayar Khosravi, Richard Socher (ICLR 2017)

Language Modeling With Tying Word Vectors

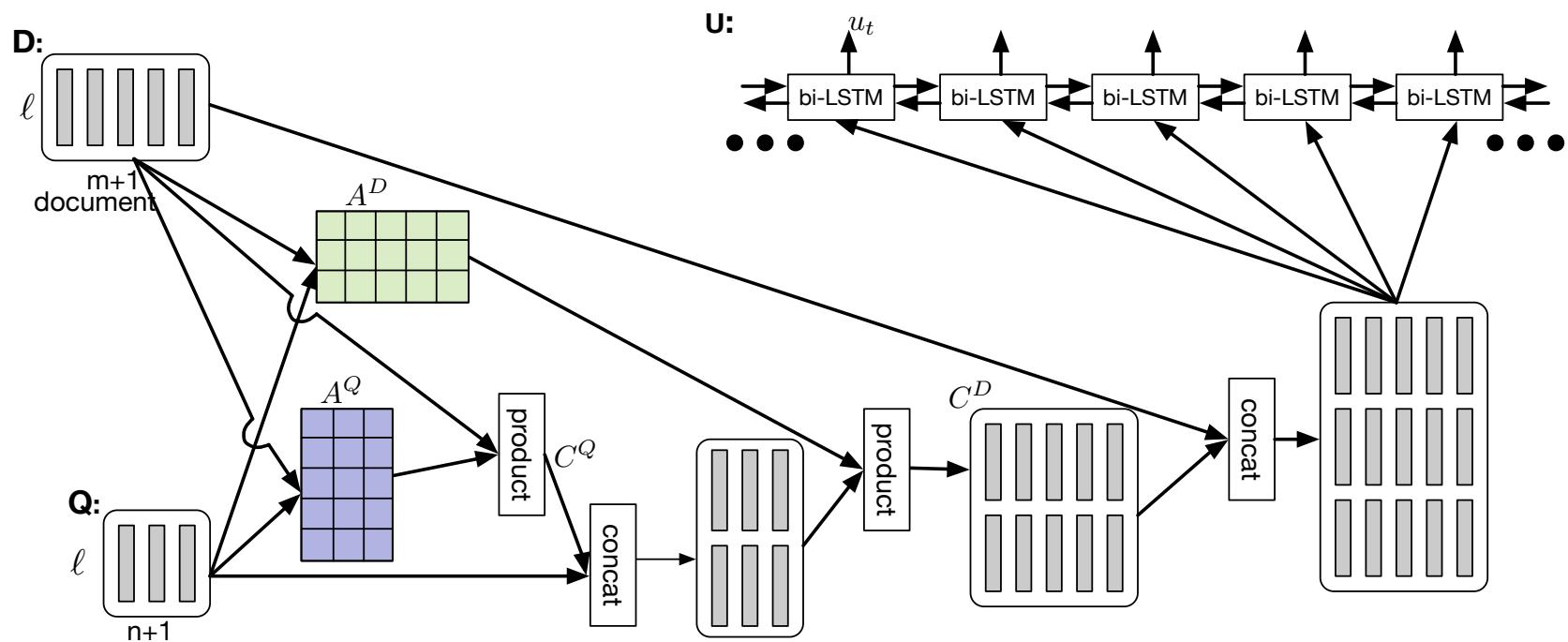
MODEL	PARAMETERS	VALIDATION	TEST
KN-5 (Mikolov & Zweig)	2M	-	141.2
KN-5 + Cache (Mikolov & Zweig)	2M	-	125.7
RNN (Mikolov & Zweig)	6M	-	124.7
RNN+LDA (Mikolov & Zweig)	7M	-	113.7
RNN+LDA+KN-5+Cache (Mikolov & Zweig)	9M	-	92.0
Deep RNN (Pascanu et al., 2013a)	6M	-	107.5
Sum-Prod Net (Cheng et al., 2014)	5M	-	100.0
LSTM (medium) (Zaremba et al., 2014)	20M	86.2	82.7
LSTM (large) (Zaremba et al., 2014)	66M	82.2	78.4
VD-LSTM (medium, untied) (Gal, 2015)	20M	81.9 ± 0.2	79.7 ± 0.1
VD-LSTM (medium, untied, MC) (Gal, 2015)	20M	-	78.6 ± 0.1
VD-LSTM (large, untied) (Gal, 2015)	66M	77.9 ± 0.3	75.2 ± 0.2
VD-LSTM (large, untied, MC) (Gal, 2015)	66M	-	73.4 ± 0.0
CharCNN (Kim et al., 2015)	19M	-	78.9
VD-RHN (Zilly et al., 2016)	32M	72.8	71.3
Pointer Sentinel-LSTM(medium) (Merity et al., 2016)	21M	72.4	70.9
38 Large LSTMs (Zaremba et al., 2014)	2.51B	71.9	68.7
10 Large VD-LSTMs (Gal, 2015)	660M	-	68.7
VD-LSTM +REAL (medium)	14M	75.7	73.2
VD-LSTM +REAL (large)	51M	71.1	68.5

Obstacle 5: Questions have input independent representations

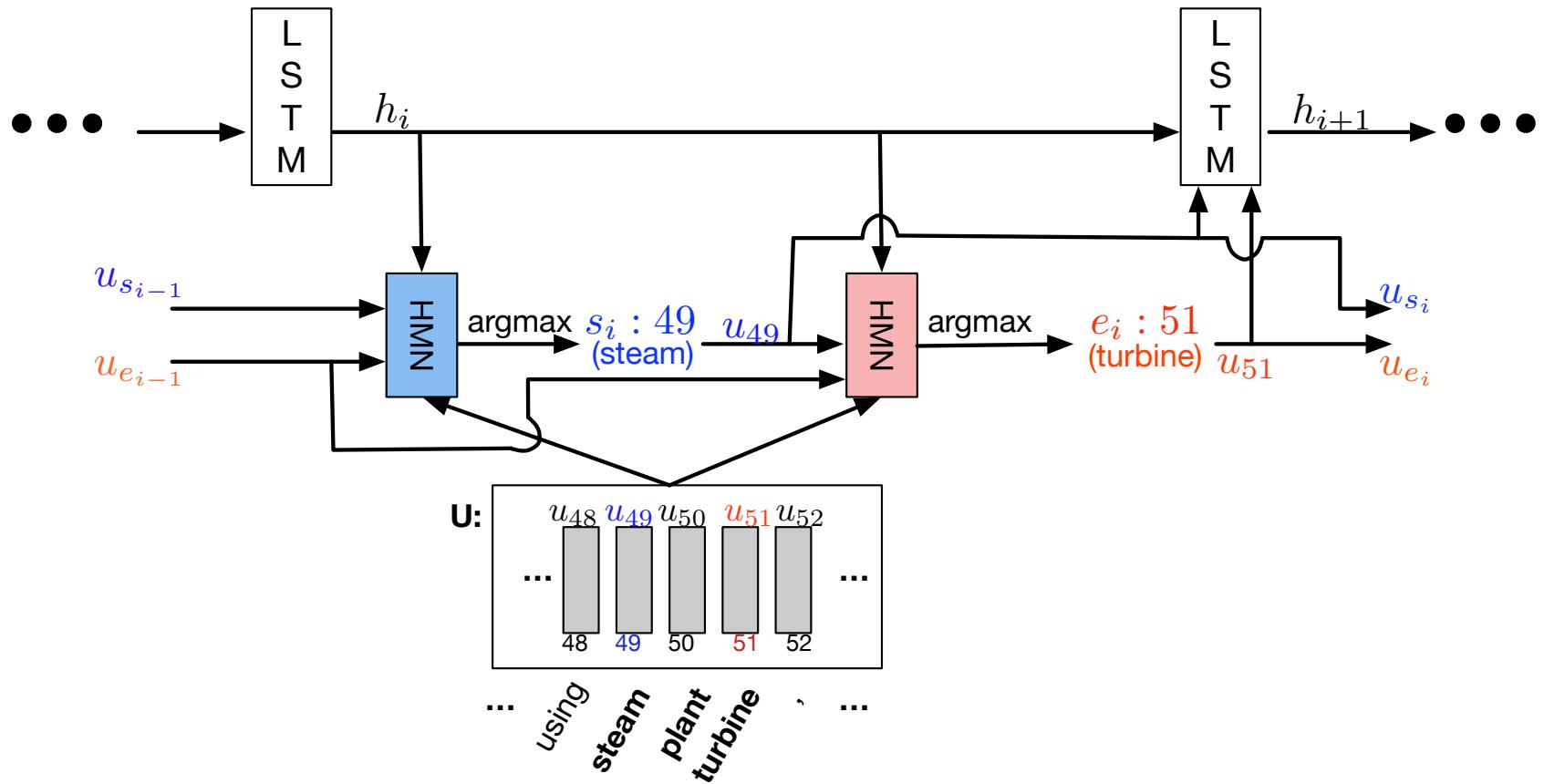
- Interdependence needed for a comprehensive QA model
- Dynamic Coattention Networks for Question Answering by Caiming Xiong, Victor Zhong, Richard Socher (ICLR 2017)



Coattention Encoder



Dynamic Decoder



Stanford Question Answering Dataset

Packet_switching

The Stanford Question Answering Dataset

Starting in the late 1950s, American computer scientist Paul Baran developed the concept Distributed Adaptive Message Block Switching with the goal to provide a fault-tolerant, efficient routing method for telecommunication messages as part of a research program at the RAND Corporation, funded by the US Department of Defense. This concept contrasted and contradicted the theretofore established principles of pre-allocation of network bandwidth, largely fortified by the development of telecommunications in the Bell System. The new concept found little resonance among network implementers until the independent work of Donald Davies at the National Physical Laboratory (United Kingdom) (NPL) in the late 1960s. Davies is credited with coining the modern name packet switching and inspiring numerous packet switching networks in Europe in the decade following, including the incorporation of the concept in the early ARPANET in the United States.

What did this concept contradict

Ground Truth Answers: This concept contrasted and contradicted the theretofore established principles of pre-allocation of network bandwidth | theretofore established principles of pre-allocation of network bandwidth | principles of pre-allocation of network bandwidth

What is Donald Davies credited with

Ground Truth Answers: Davies is credited with coining the modern name packet switching and inspiring numerous packet switching networks in Europe | coining the modern name packet switching and inspiring numerous packet switching networks | coining the modern name packet switching

What did Paul Baran develop in the late 1950's

Ground Truth Answers: the concept Distributed Adaptive Message Block Switching | the concept Distributed Adaptive Message Block

Results on SQuAD Competition

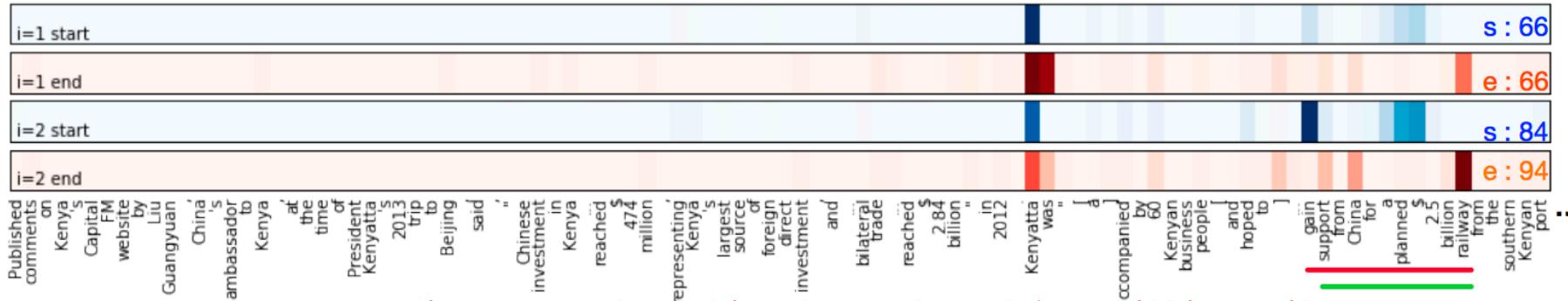
Model	Dev EM	Dev F1	Test EM	Test F1
<i>Ensemble</i>				
DCN (Ours)	70.3	79.4	71.2	80.4
Microsoft Research Asia *	—	—	69.4	78.3
Allen Institute *	69.2	77.8	69.9	78.1
Singapore Management University *	67.6	76.8	67.9	77.0
Google NYC *	68.2	76.7	—	—
<i>Single model</i>				
DCN (Ours)	65.4	75.6	66.2	75.9
Microsoft Research Asia *	65.9	75.2	65.5	75.0
Google NYC *	66.4	74.9	—	—
Singapore Management University *	—	—	64.7	73.7
Carnegie Mellon University *	—	—	62.5	73.3
Dynamic Chunk Reader (Yu et al., 2016)	62.5	71.2	62.5	71.0
Match-LSTM (Wang & Jiang, 2016)	59.1	70.0	59.5	70.3
Baseline (Rajpurkar et al., 2016)	40.0	51.0	40.4	51.0
Human (Rajpurkar et al., 2016)	81.4	91.0	82.3	91.2

Results are at time of ICLR submission

See <https://rajpurkar.github.io/SQuAD-explorer/> for latest results

Dynamic Decoder Visualization

Question 2: What did the Kenyan business people hope for when meeting with the Chinese?



Answer: gain support from China for a planned \$2.5 billion railway

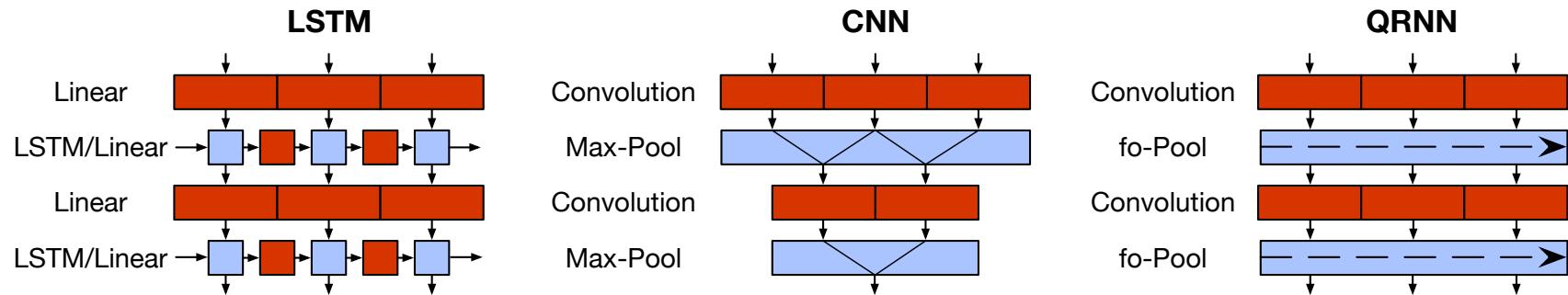
Groundtruth: support from China for a planned \$2.5 billion railway

Obstacle 6: RNNs are Slow

- RNNs are the basic building block for deepNLP
- Idea: Take the best and parallelizable parts of RNNs and CNNs
- Quasi-Recurrent Neural Networks by James Bradbury, Stephen Merity, Caiming Xiong & Richard Socher (ICLR 2017)

Quasi-Recurrent Neural Network

- Combines best of both model families



- Convolutions for parallelism across time:

$$\mathbf{z}_t = \tanh(\mathbf{W}_z^1 \mathbf{x}_{t-1} + \mathbf{W}_z^2 \mathbf{x}_t)$$

$$\mathbf{f}_t = \sigma(\mathbf{W}_f^1 \mathbf{x}_{t-1} + \mathbf{W}_f^2 \mathbf{x}_t)$$

$$\mathbf{o}_t = \sigma(\mathbf{W}_o^1 \mathbf{x}_{t-1} + \mathbf{W}_o^2 \mathbf{x}_t).$$



$$\mathbf{Z} = \tanh(\mathbf{W}_z * \mathbf{X})$$

$$\mathbf{F} = \sigma(\mathbf{W}_f * \mathbf{X})$$

$$\mathbf{O} = \sigma(\mathbf{W}_o * \mathbf{X}),$$

- Element-wise gated recurrence for parallelism across channels: $\mathbf{h}_t = \mathbf{f}_t \odot \mathbf{h}_{t-1} + (1 - \mathbf{f}_t) \odot \mathbf{z}_t,$

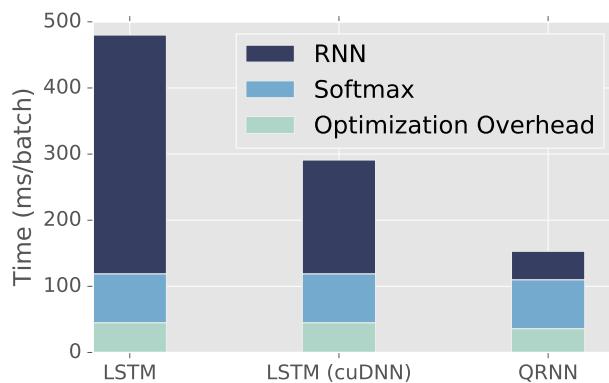
Q-RNN Experiments: Language Modeling

- Paper by James Bradbury, Stephen Merity, Caiming Xiong, Richard Socher (ICLR 2017)

- Better

Model	Parameters	Validation	Test
LSTM (medium) (Zaremba et al., 2014)	20M	86.2	82.7
Variational LSTM (medium) (Gal & Ghahramani, 2016)	20M	81.9	79.7
LSTM with CharCNN embeddings (Kim et al., 2016)	19M	—	78.9
Zoneout + Variational LSTM (medium) (Merity et al., 2016)	20M	84.4	80.6
<i>Our models</i>			
LSTM (medium)	20M	85.7	82.0
QRNN (medium)	18M	82.9	79.9
QRNN + zoneout ($p = 0.1$) (medium)	18M	82.1	78.3

- Faster



Batch size	Sequence length				
	32	64	128	256	512
8	5.5x	8.8x	11.0x	12.4x	16.9x
16	5.5x	6.7x	7.8x	8.3x	10.8x
32	4.2x	4.5x	4.9x	4.9x	6.4x
64	3.0x	3.0x	3.0x	3.0x	3.7x
128	2.1x	1.9x	2.0x	2.0x	2.4x
256	1.4x	1.4x	1.3x	1.3x	1.3x

Q-RNN Experiments: Sentiment Analysis

- Better and faster than LSTMs

- Interpretable

- Example:

- Initial positive review

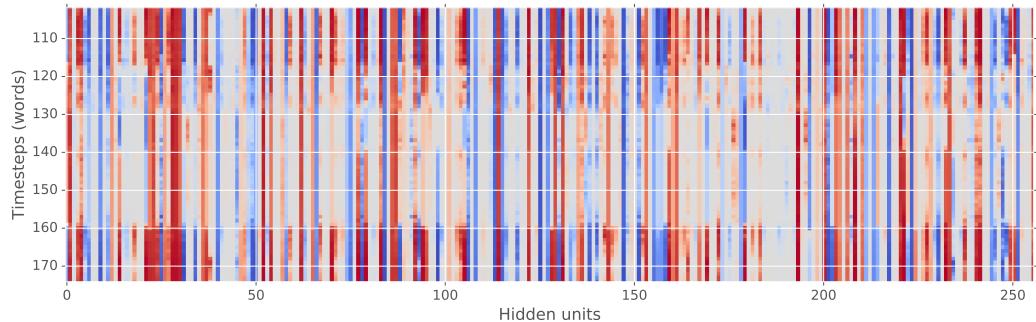
- *Review starts out positive*

At 117: “not exactly a bad story”

At 158: “I recommend this movie to everyone, even if you’ve never played the game”

- <https://metamind.io/research/new-neural-network-building-block-allows-faster-and-more-accurate-text-understanding/>

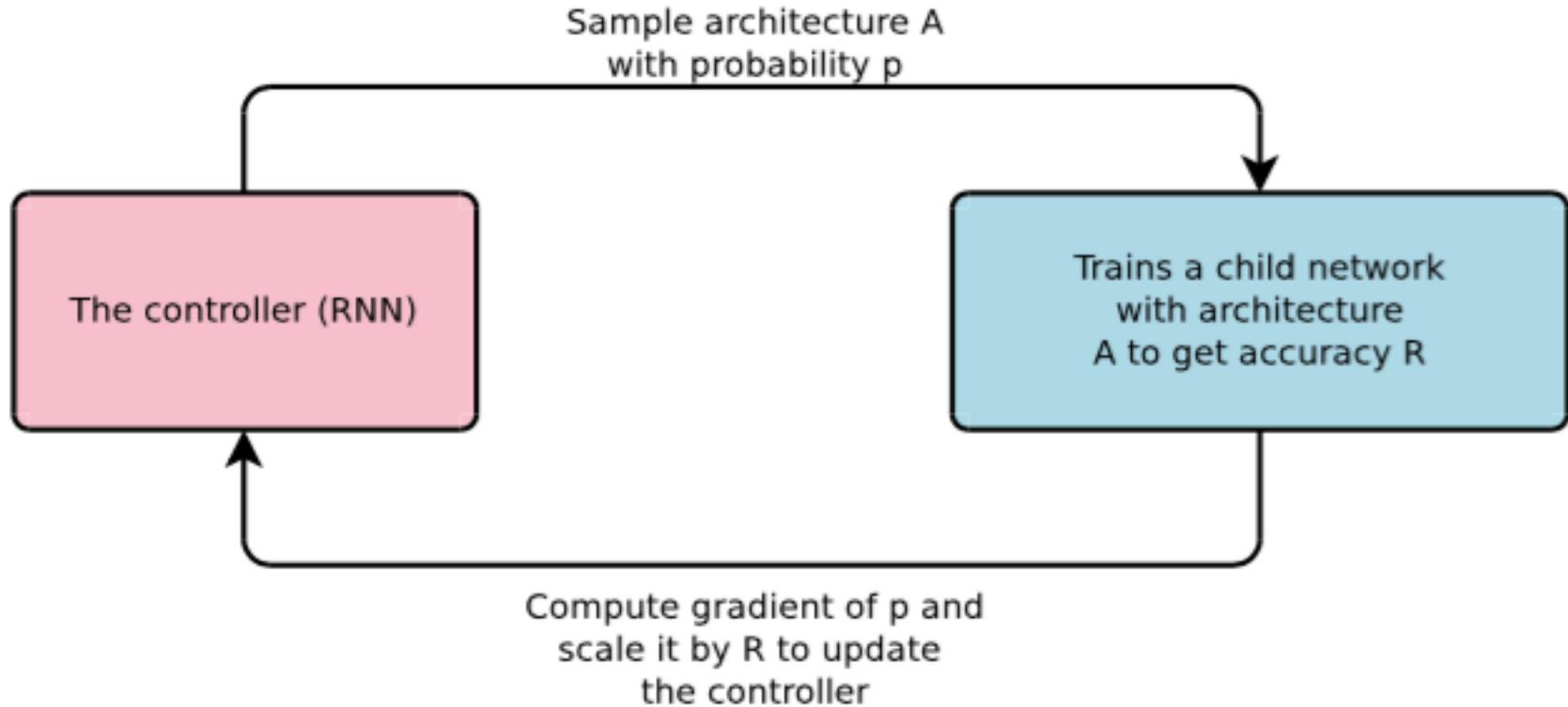
Model	Time / Epoch (s)	Test Acc (%)
BSVM-bi (Wang & Manning, 2012)	—	91.2
2 layer sequential BoW CNN (Johnson & Zhang, 2014)	—	92.3
Ensemble of RNNs and NB-SVM (Mesnil et al., 2014)	—	92.6
2-layer LSTM (Longpre et al., 2016)	—	87.6
Residual 2-layer bi-LSTM (Longpre et al., 2016)	—	90.1
<i>Our models</i>		
Deeply connected 4-layer LSTM (cuDNN optimized)	480	90.9
Deeply connected 4-layer QRNN	150	91.4
D.C. 4-layer QRNN with $k = 4$	160	91.1



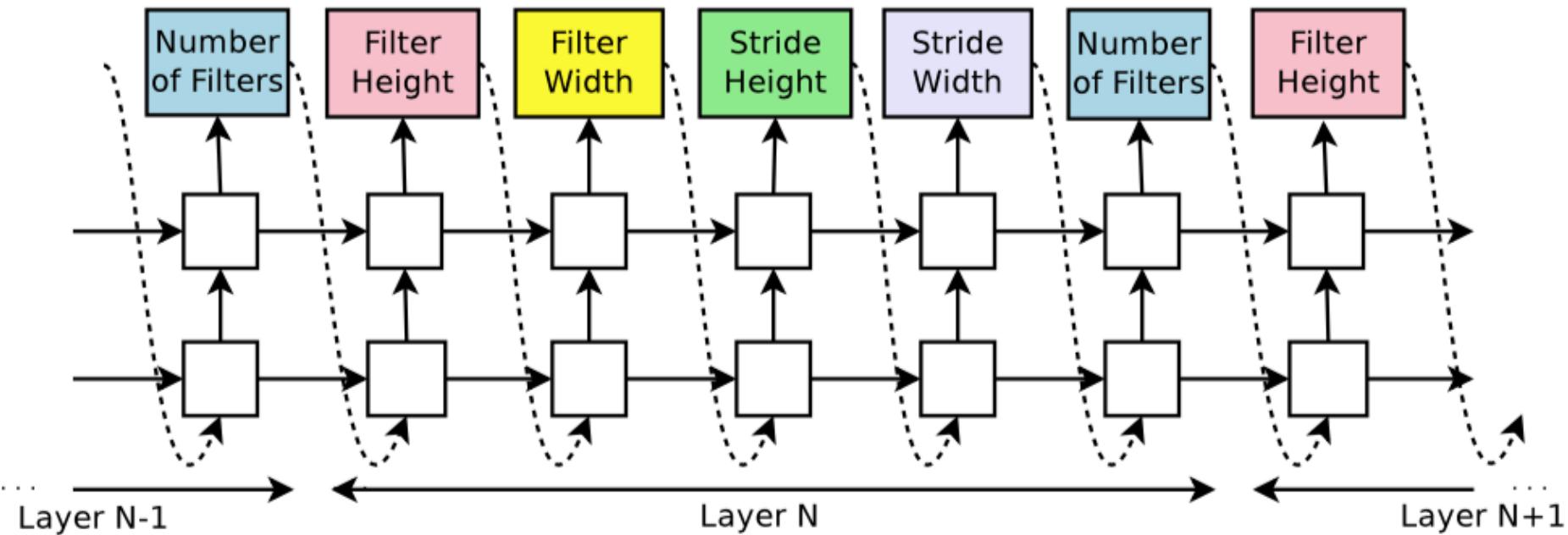
Obstacle 7: Architecture Search is Slow

- Manual process that requires a lot of expertise
- What if we could use AI to find the right architecture for any problem?
- Neural architecture search with reinforcement learning by Zoph and Le, 2016

Neural Architecture Search



Example: CNN Controller

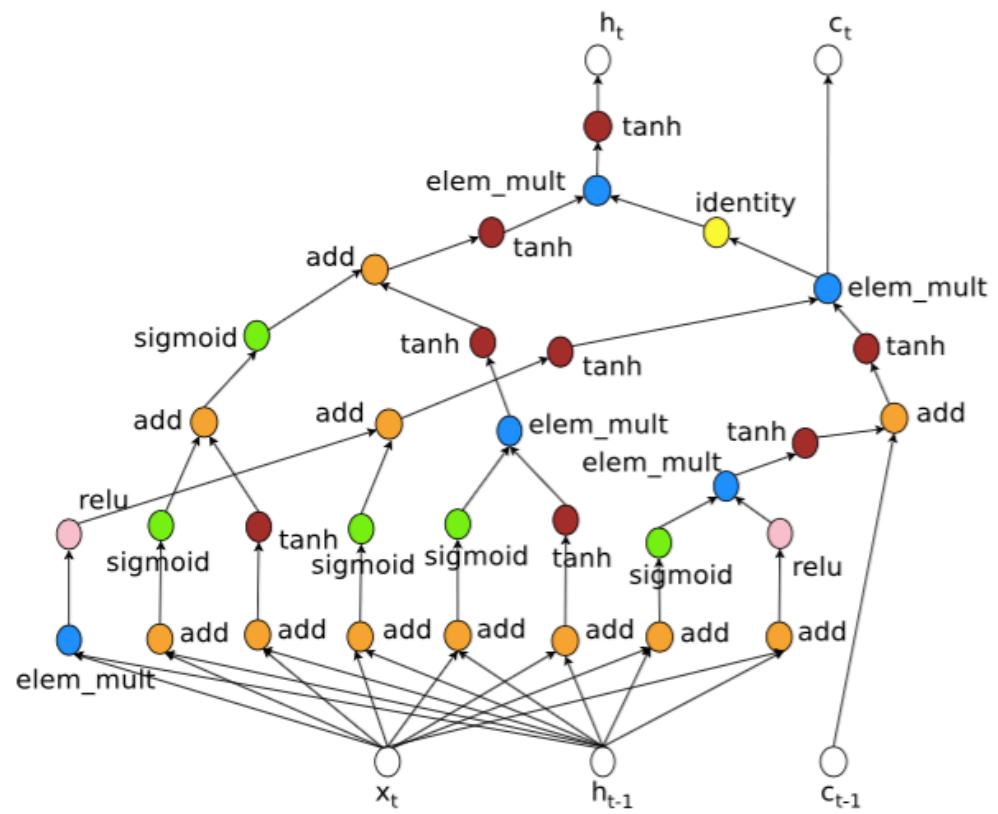
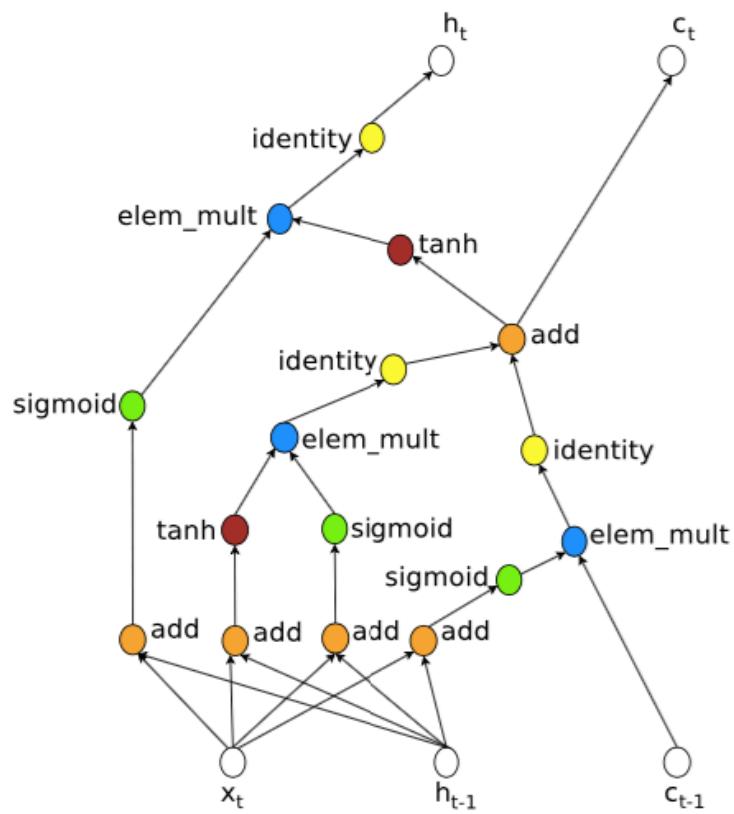


Used Reinforcement Learning to train the RNN Controller

Amazing Perplexity Reduction for Language Modeling

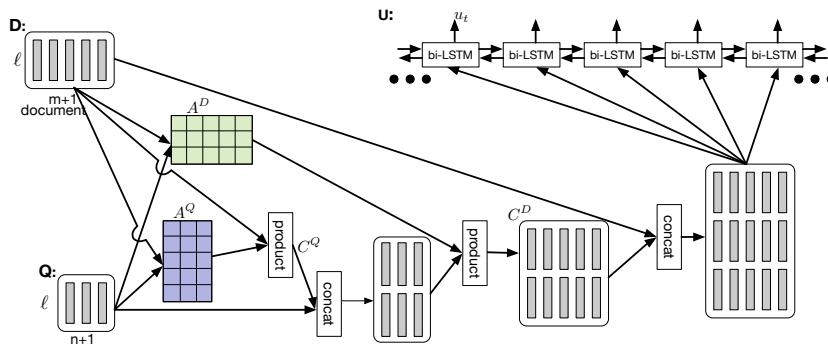
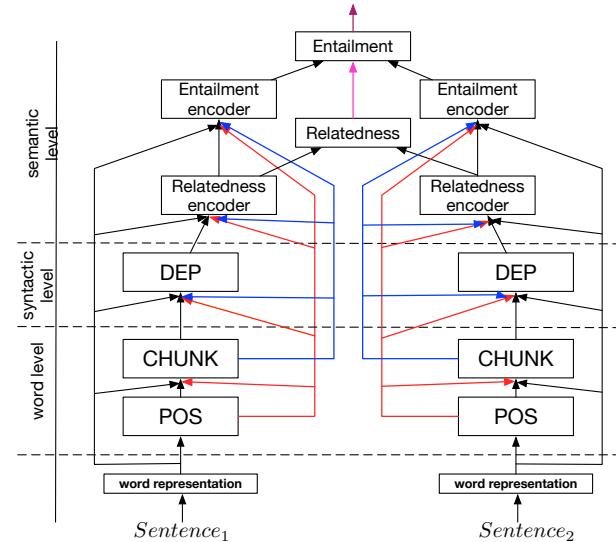
Model	Parameters	Test Perplexity
Mikolov & Zweig (2012) - KN-5	2M [‡]	141.2
Mikolov & Zweig (2012) - KN5 + cache	2M [‡]	125.7
Mikolov & Zweig (2012) - RNN	6M [‡]	124.7
Mikolov & Zweig (2012) - RNN-LDA	7M [‡]	113.7
Mikolov & Zweig (2012) - RNN-LDA + KN-5 + cache	9M [‡]	92.0
Pascanu et al. (2013) - Deep RNN	6M	107.5
Cheng et al. (2014) - Sum-Prod Net	5M [‡]	100.0
Zaremba et al. (2014) - LSTM (medium)	20M	82.7
Zaremba et al. (2014) - LSTM (large)	66M	78.4
Gal (2015) - Variational LSTM (medium, untied)	20M	79.7
Gal (2015) - Variational LSTM (medium, untied, MC)	20M	78.6
Gal (2015) - Variational LSTM (large, untied)	66M	75.2
Gal (2015) - Variational LSTM (large, untied, MC)	66M	73.4
Kim et al. (2015) - CharCNN	19M	78.9
Press & Wolf (2016) - Variational LSTM, shared embeddings	51M	73.2
Merity et al. (2016) - Zoneout + Variational LSTM (medium)	20M	80.6
Merity et al. (2016) - Pointer Sentinel-LSTM (medium)	21M	70.9
Inan et al. (2016) - VD-LSTM + REAL (large)	51M	68.5
Zilly et al. (2016) - Variational RHN, shared embeddings	24M	66.0
Neural Architecture Search with base 8	32M	67.9
Neural Architecture Search with base 8 and shared embeddings	25M	64.0
Neural Architecture Search with base 8 and shared embeddings	54M	62.4

LSTM Cell vs NAS Cell



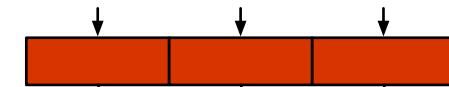
Lots of Limits for deepNLP

- Q&A
- Multitask learning
- Combined multimodal, logical and memory-based reasoning
- Learning from few examples

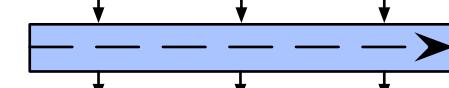


QRNN

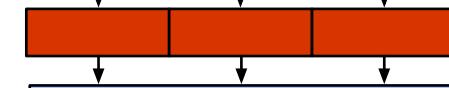
Convolution



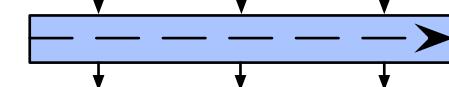
fo-Pool



Convolution



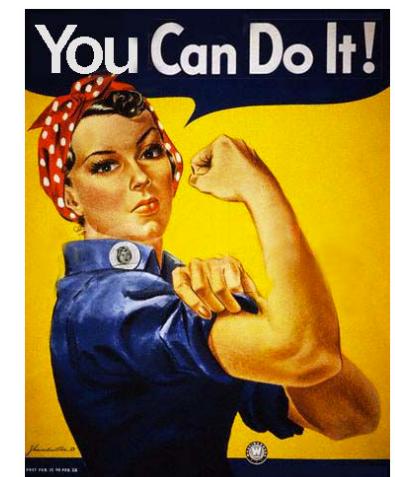
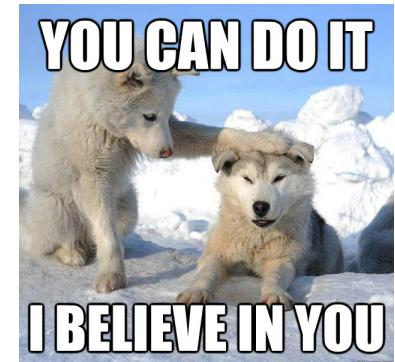
fo-Pool



DeepNLP

Congratulations!

Good luck with the
projects and PA4s



Framework for Tackling NLP

A joint model
for
comprehensive
QA

QA Examples

I: Mary walked to the bathroom.

I: Sandra went to the garden.

I: Daniel went back to the garden.

I: Sandra took the milk there.

Q: Where is the milk?

A: garden

I: Everybody is happy.

Q: What's the sentiment?

A: positive

I: I think this model is incredible

Q: In French?

A: Je pense que ce modèle est incroyable.

I:



Q: What color are the bananas?

A: Green.

Move from $\{x_i, y_i\}$ to $\{x_i, q_i, y_i\}$