Natural Language Processing with Deep Learning

CS224N/Ling284

Richard Socher

Lecture 15: Model Overview and Memory Networks

Outline

- Last minute tips for projects
- Model overview and combinations
- Dynamic memory networks

Last minute tips

- Nothing works and everything is too slow → Panic

- Simplify model → Go back to basics: bag of vectors + nnet
- Make a smaller network and dataset for debugging
- Once no bugs: increase model size
- Make sure you can overfit to your dataset
- Plot your training and dev errors over training iterations
- Then regularize with L2 and Dropout
- Then do hyperparameter search

- Come to OH! (

Model comparison

- **Bag of Vectors**: Surprisingly good baseline for simple text classification problems. Especially if followed by a few relu layers!
- **Window Model**: Good for single word classification for problems that do not need wide context, e.g. POS
- **CNNs:** good for classification, unclear how to incorporate phrase level annotation (can only take a single label), need zero padding for shorter phrases, hard to interpret, easy to parallelize on GPUs, can be very efficient and versatile
- **Recurrent Neural Networks**: Cognitively plausible (reading from left to right, keeping a state), not best for classification (n-gram), slower than CNNs, can do sequence tagging and classification, very active research, amazing with attention mechanisms
- **TreeRNNs**: Linguistically plausible, hard to parallelize, tree structures are discrete and harder to optimize, need a parser
- **Combinations and extensions!**

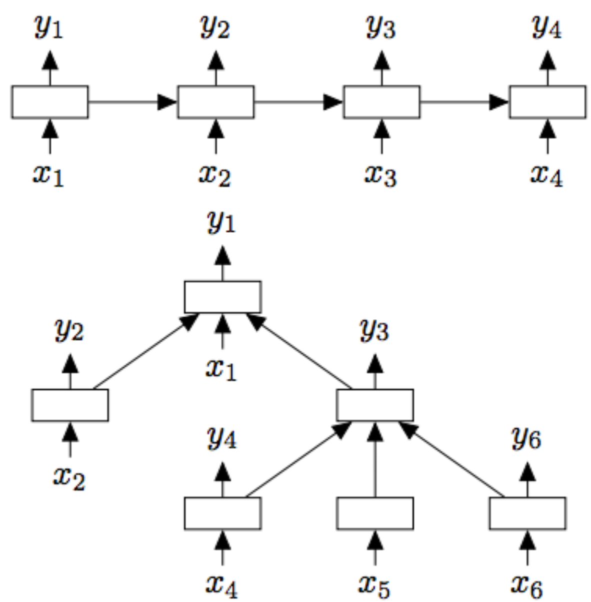
But, there's more

- Combine and extend creatively

- Rarely do we use the vanilla models as is

TreeLSTMs

- LSTMs are great
- TreeRNNs can benefit from gates too → TreeRNNs + LSTMs
- Improved Semantic Representations From Tree-Structured Long Short-Term Memory Networks by Kai Sheng Tai, Richard Socher, Christopher D. Manning



TreeLSTMs

- Standard LSTM
- Only has one child

$$i_t = \sigma\left(W^{(i)}x_t + U^{(i)}h_{t-1} + b^{(i)}\right),$$
$$f_t = \sigma\left(W^{(f)}x_t + U^{(f)}h_{t-1} + b^{(f)}\right),$$
$$o_t = \sigma\left(W^{(o)}x_t + U^{(o)}h_{t-1} + b^{(o)}\right),$$
$$u_t = \tanh\left(W^{(u)}x_t + U^{(u)}h_{t-1} + b^{(u)}\right),$$
$$c_t = i_t \odot u_t + f_t \odot c_{t-1},$$
$$h_t = o_t \odot \tanh(c_t),$$

TreeLSTM

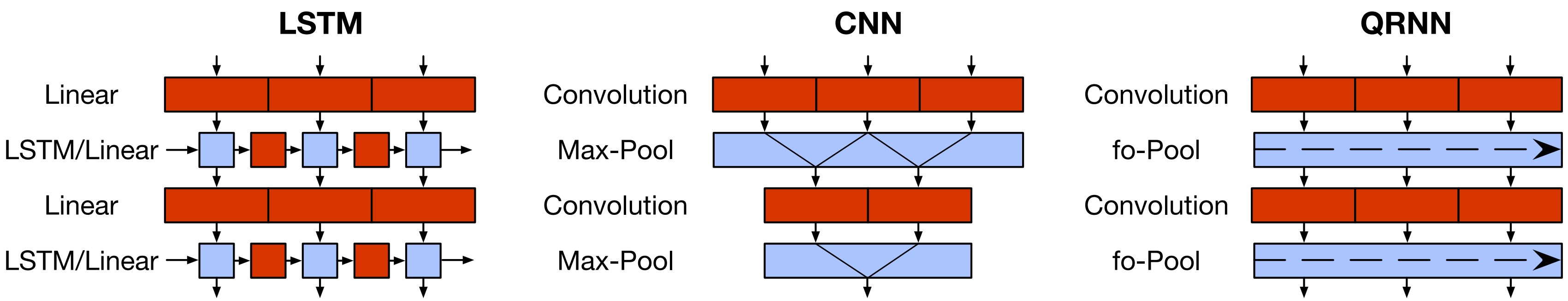
Has multiple child nodes:

$$\tilde{h}_j = \sum_{k \in C(j)} h_k,$$
$$i_j = \sigma\left(W^{(i)}x_j + U^{(i)}\tilde{h}_j + b^{(i)}\right),$$
$$f_{jk} = \sigma\left(W^{(f)}x_j + U^{(f)}h_k + b^{(f)}\right),$$
$$o_j = \sigma\left(W^{(o)}x_j + U^{(o)}\tilde{h}_j + b^{(o)}\right),$$
$$u_j = \tanh\left(W^{(u)}x_j + U^{(u)}\tilde{h}_j + b^{(u)}\right),$$
$$c_j = i_j \odot u_j + \sum_{k \in C(j)} f_{jk} \odot c_k,$$
$$h_j = o_j \odot \tanh(c_j),$$

RNNs are Slow → Combine with CNNs

- RNNs are the most common 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

Quasi-Recurrent Neural Network



- Parallelism computation 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:

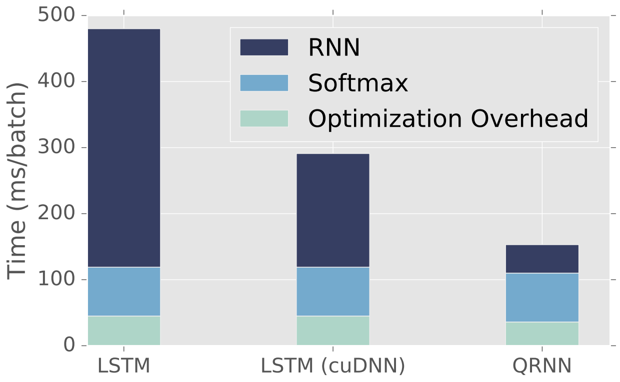
$$\mathbf{h}_t = \mathbf{f}_t \odot \mathbf{h}_{t-1} + (1 - \mathbf{f}_t) \odot \mathbf{z}_t,$$

Q-RNNs for Language Modeling

- 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
Our models			
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



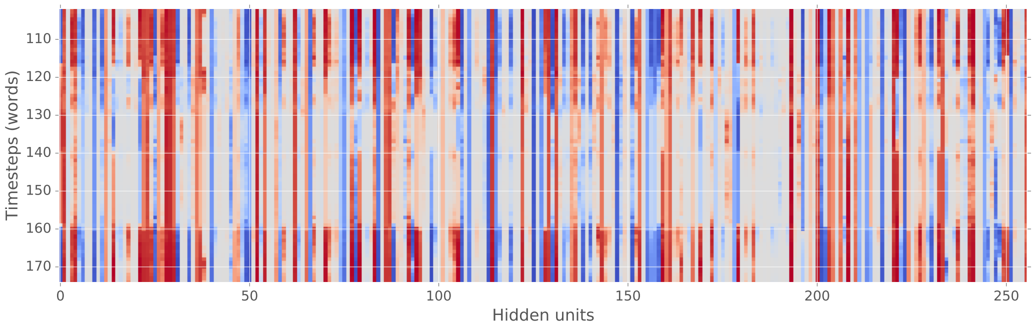
		Sequence length				
		32	64	128	256	512
Batch size	8	**5.5x**	**8.8x**	**11.0x**	**12.4x**	**16.9x**
	16	**5.5x**	**6.7x**	**7.8x**	**8.3x**	**10.8x**
	32	**4.2x**	**4.5x**	**4.9x**	**4.9x**	**6.4x**
	64	**3.0x**	**3.0x**	**3.0x**	**3.0x**	**3.7x**
	128	**2.1x**	**1.9x**	**2.0x**	**2.0x**	**2.4x**
	256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

Q-RNNs for Sentiment Analysis

- Often better and faster than LSTMs

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
Our models		
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

- More interpretable

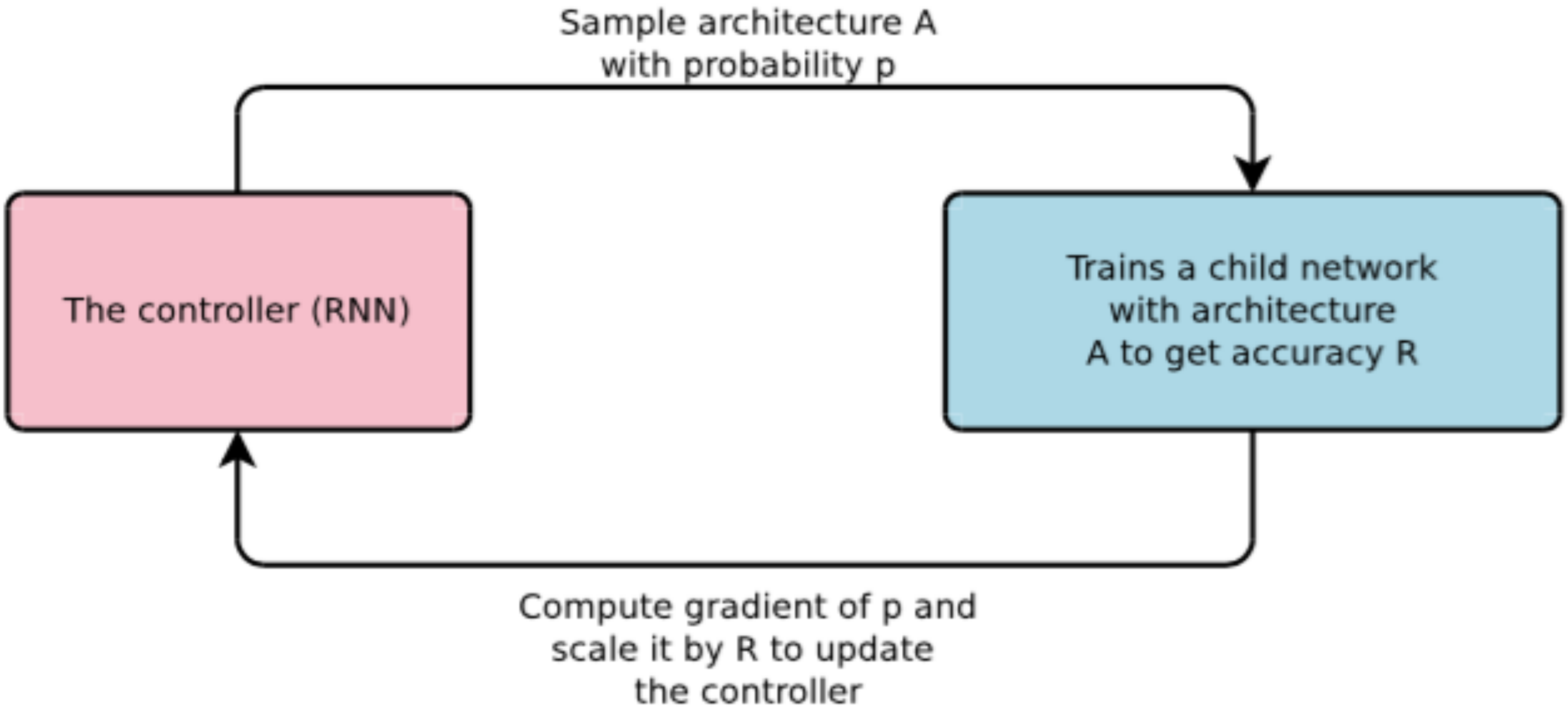


- Example:
- Initial positive review
- *Review starts out positive*
  At 117: "not exactly a bad story"
  At 158: "I recommend this movie to everyone, even if you've never played the game"

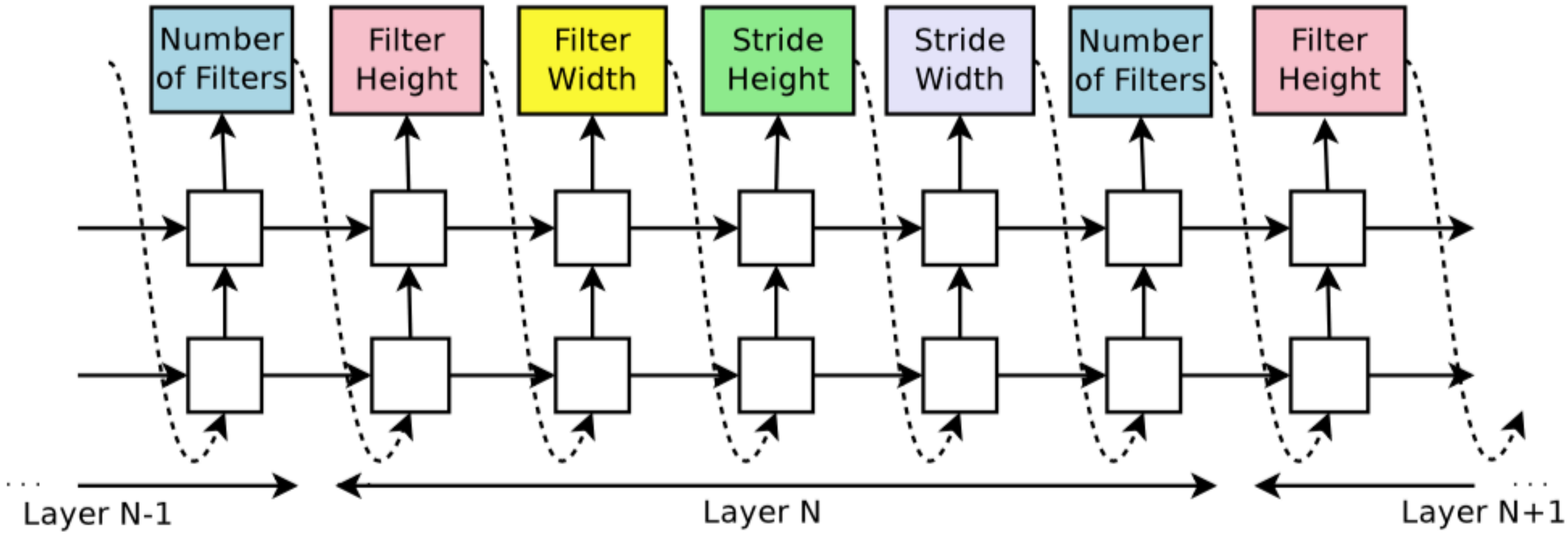
Neural Architecture Search!

- Manual process of finding best units 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

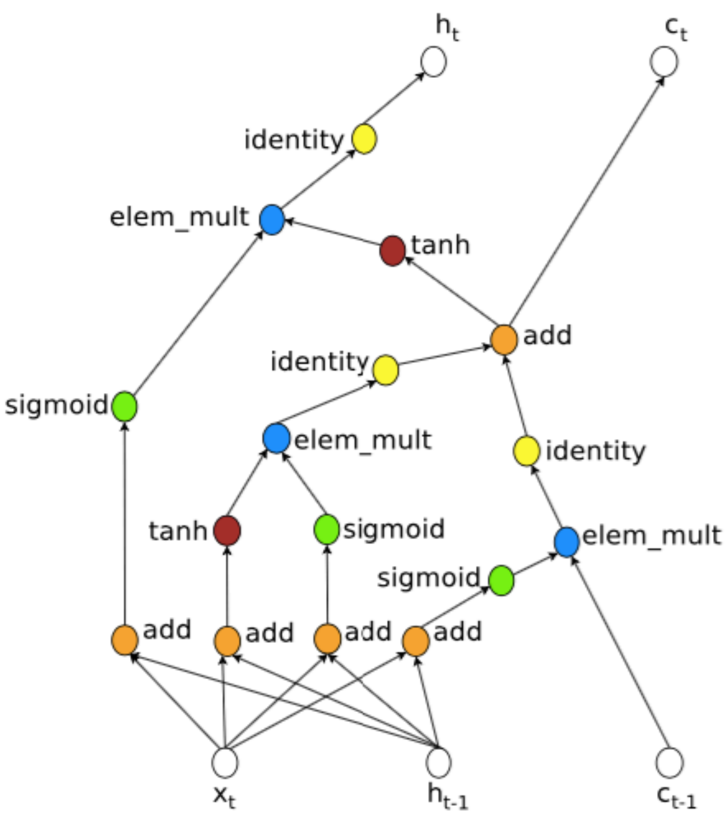


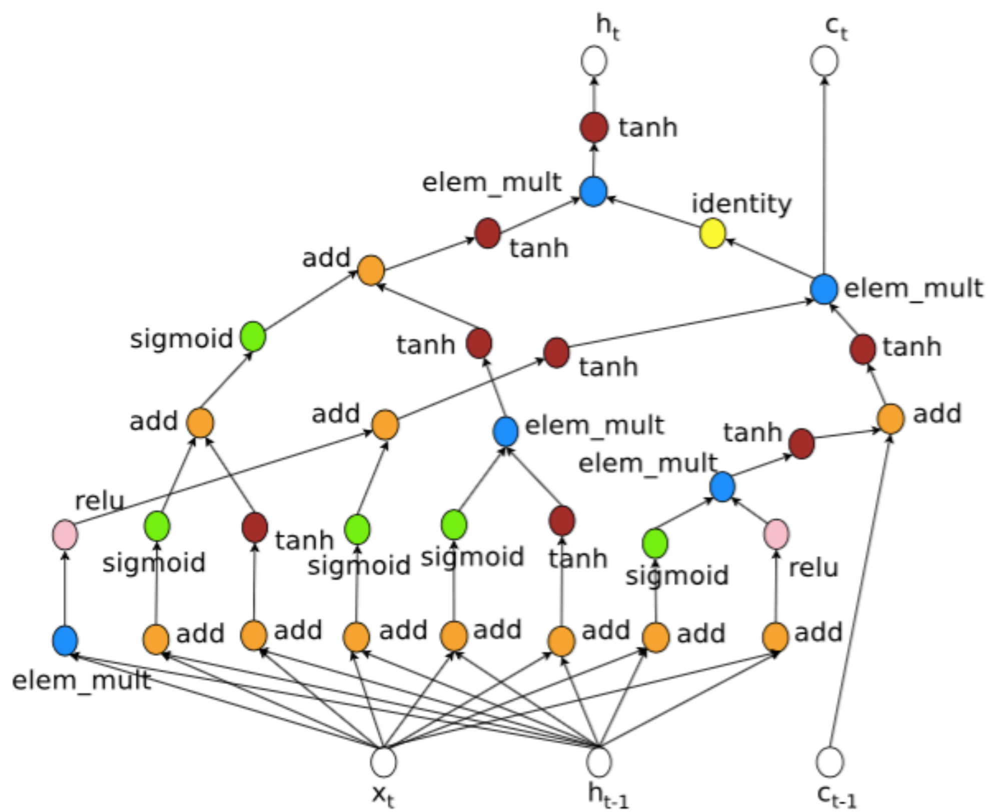
Example: CNN Controller



Used Reinforcement Learning to train the RNN Controller

LSTM Cell vs NAS Cell





Nice 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

More complex tasks need more complex architectures

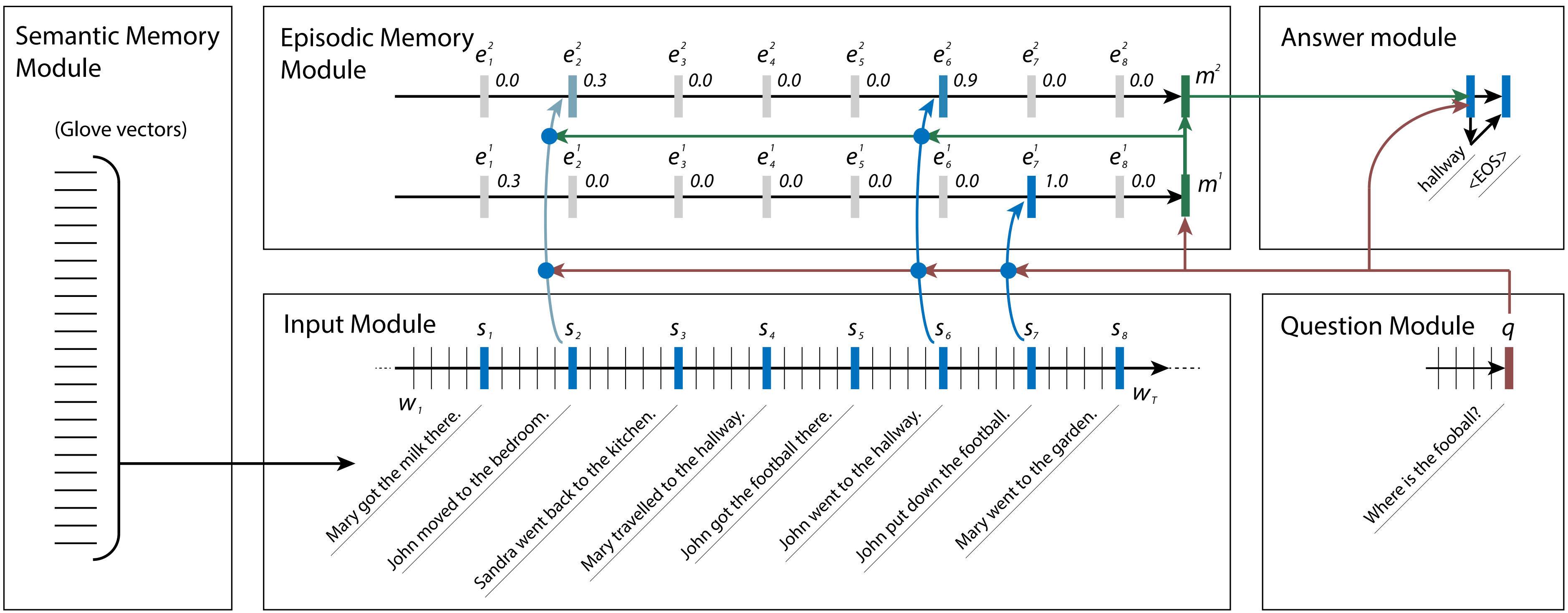
- So far, we looked at basic sequence models and seq2seq models
- As you know from the default final project, some tasks require more complex **memory components**
- One of the first ones that was shown to work on both synthetic problems and real NLP tasks:
- Dynamic Memory Networks by
  Ankit Kumar, Ozan Irsoy, Peter Ondruska, Mohit Iyyer, James Bradbury, Ishaan Gulrajani, Victor Zhong, Romain Paulus, Richard Socher

High level idea for harder questions

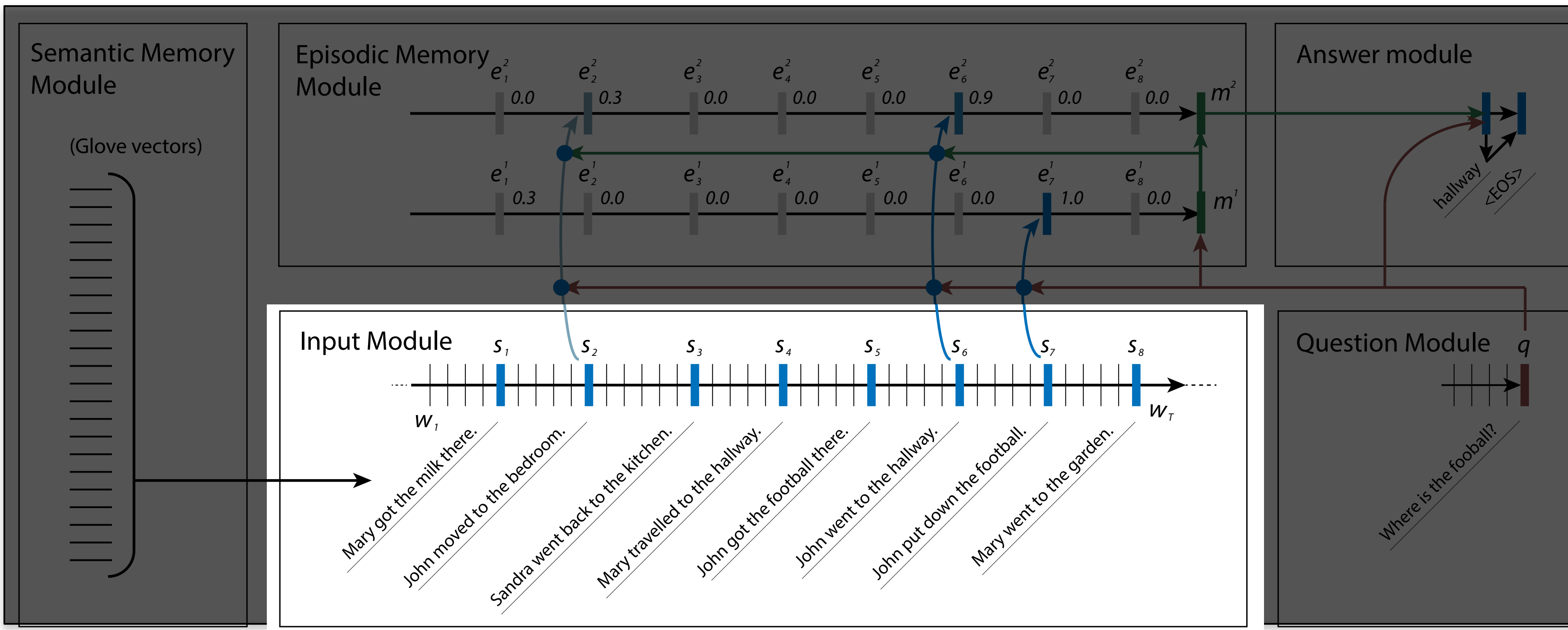
- Imagine having to read an article, memorize it, then get asked various questions → Hard!
- You can't store everything in working memory
- **Optimal:** give you the input data, give you the question, allow as many glances as possible

```
1 Mary moved to the bathroom.
2 John went to the hallway.
3 Where is Mary?         bathroom         1
4 Daniel went back to the hallway.
5 Sandra moved to the garden.
6 Where is Daniel?       hallway          4
7 John moved to the office.
8 Sandra journeyed to the bathroom.
9 Where is Daniel?       hallway          4
10 Mary moved to the hallway.
11 Daniel travelled to the office.
12 Where is Daniel?      office           11
13 John went back to the garden.
14 John moved to the bedroom.
15 Where is Sandra?      bathroom         8
1 Sandra travelled to the office.
2 Sandra went to the bathroom.
3 Where is Sandra?       bathroom         2
```

Dynamic Memory Network

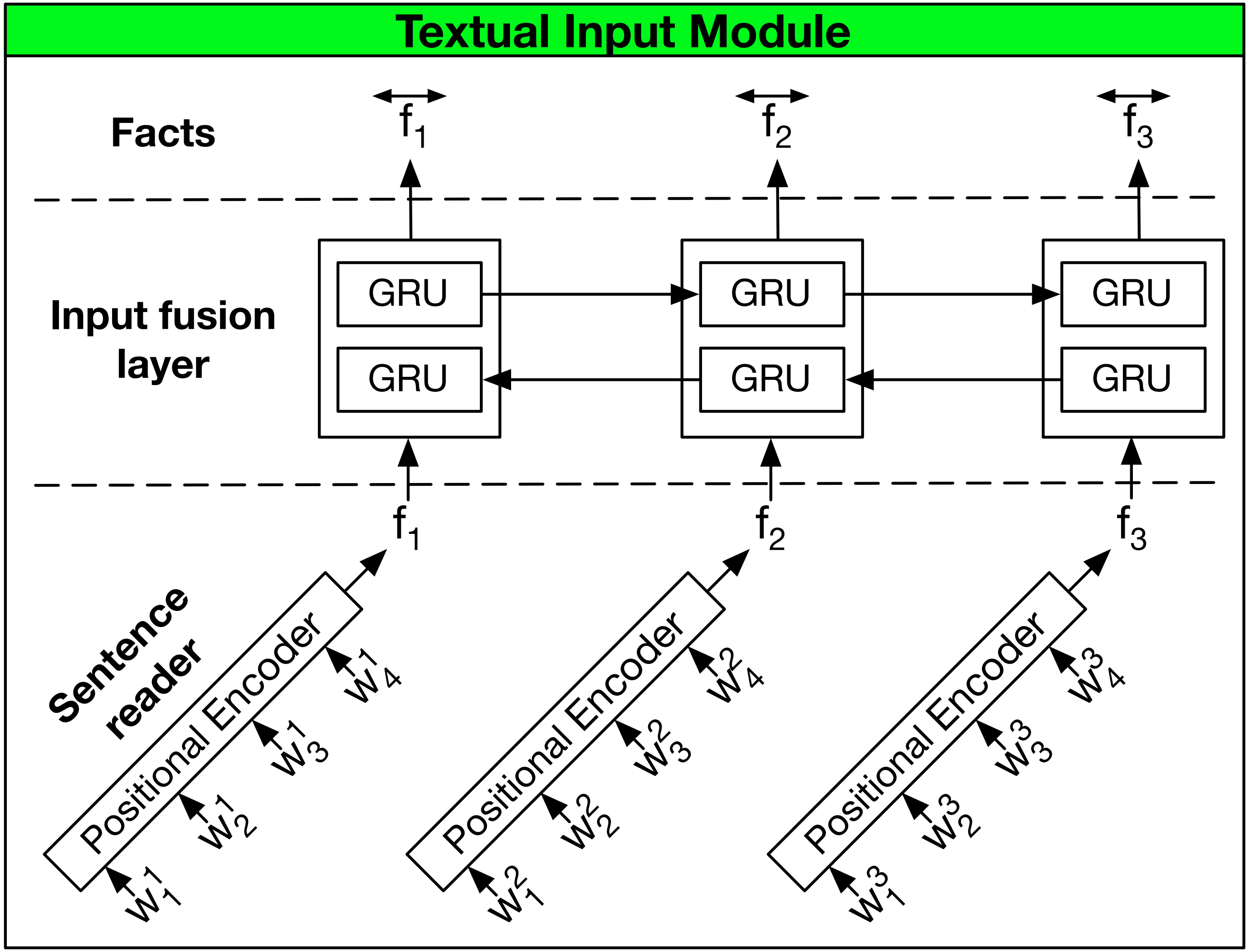


The Modules: Input

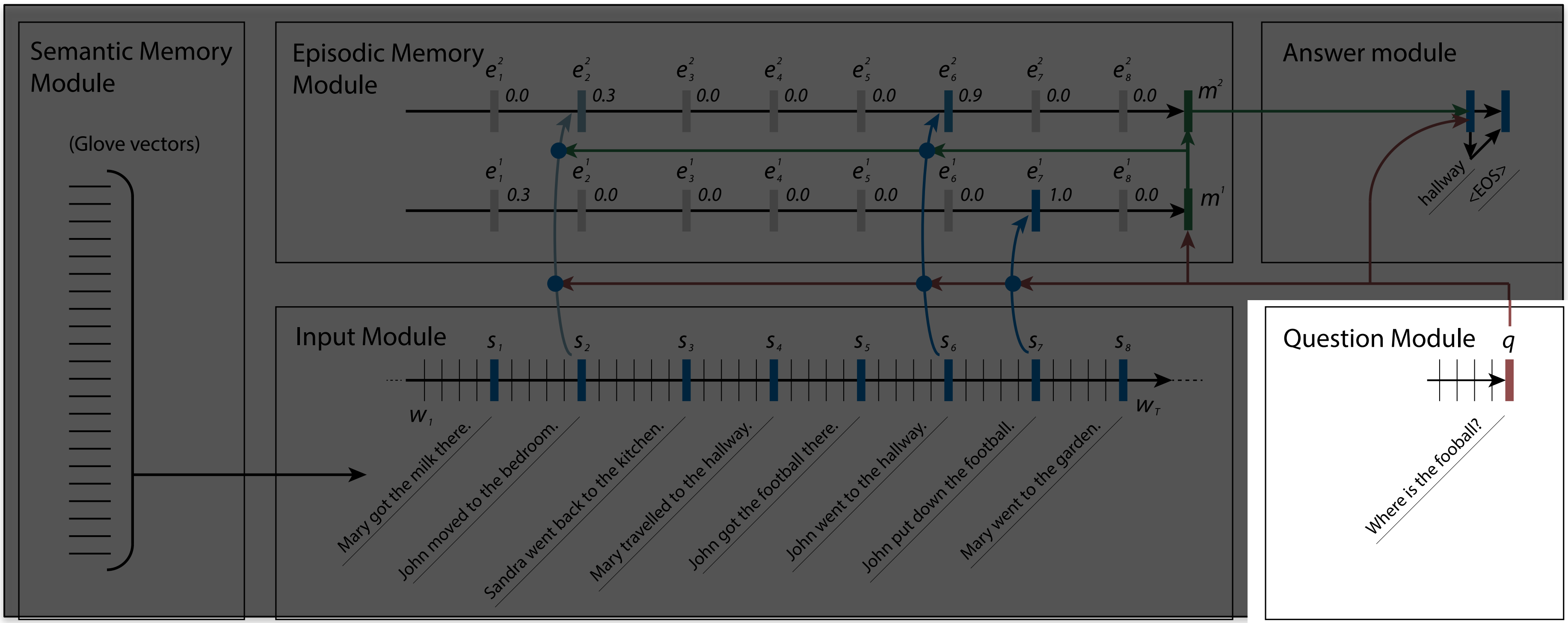


Standard GRU. The last hidden state of each sentence is accessible.

Further Improvement: BiGRU

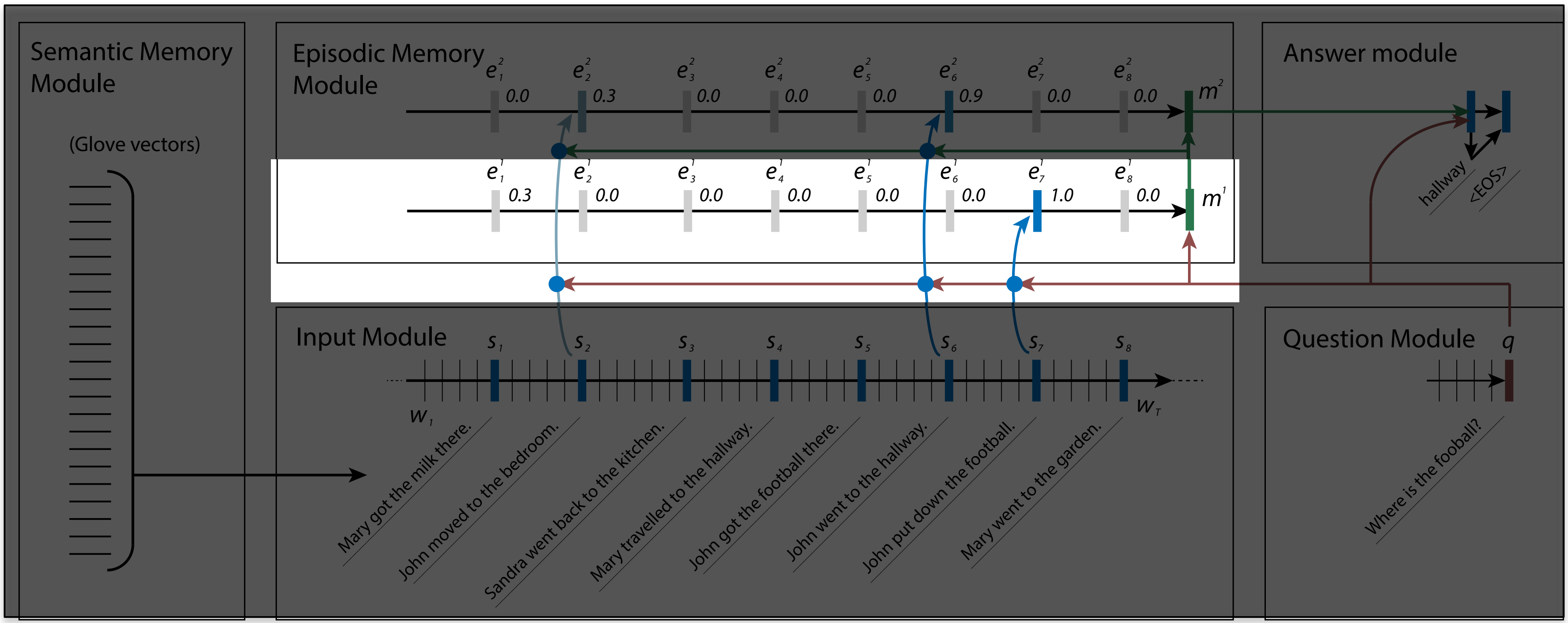


The Modules: Question



$$q_t = GRU(v_t, q_{t-1}),$$

The Modules: Episodic Memory



$$h_i^t = g_i^t GRU(s_i, h_{i-1}^t) + (1 - g_i^t) h_{i-1}^t$$

Last hidden state: m^t

The Modules: Episodic Memory

- Gates are activated if sentence relevant to the question or memory

$$z_i^t = [s_i \circ q \,; s_i \circ m^{t-1}; |s_i - q| \,; |s_i - m^{t-1}|]$$

$$Z_i^t = W^{(2)} \tanh\left(W^{(1)} z_i^t + b^{(1)}\right) + b^{(2)}$$

- When

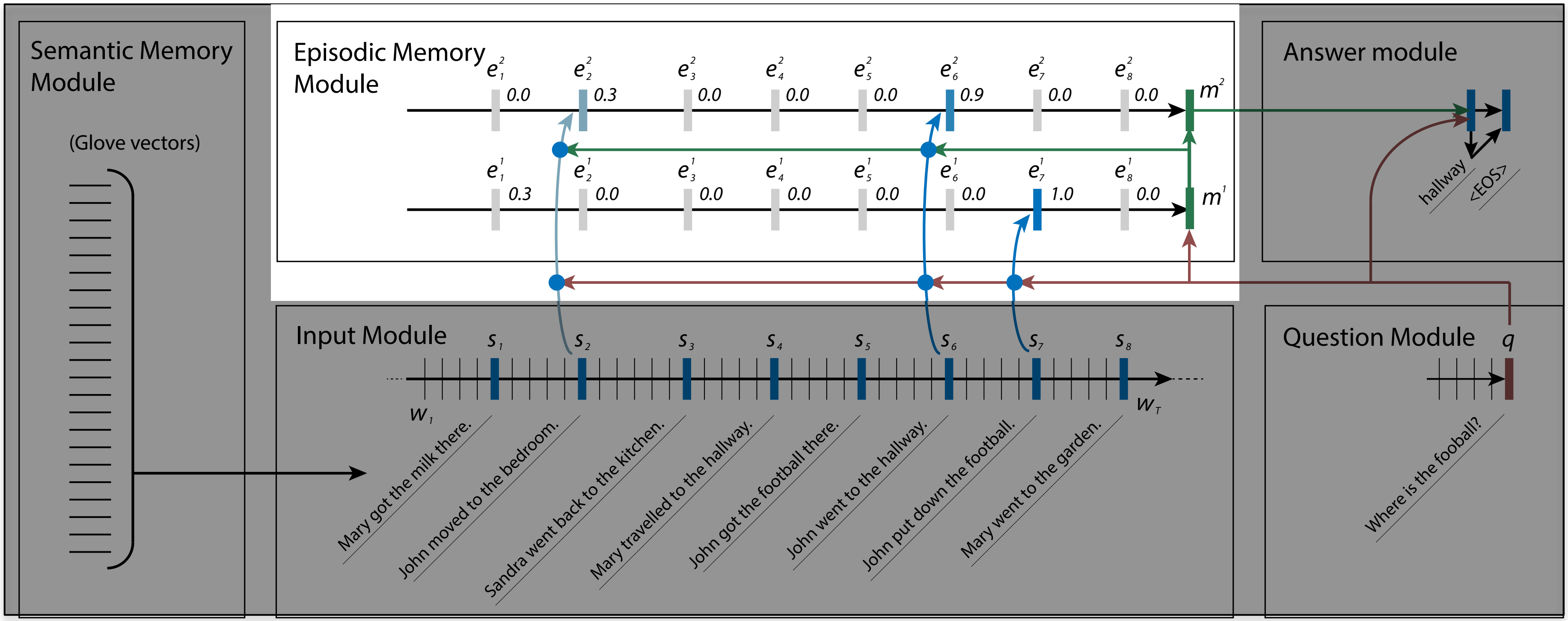
$$g_i^t = \frac{\exp(Z_i^t)}{\sum_{k=1}^{M_i} \exp(Z_k^t)}$$

e

summ

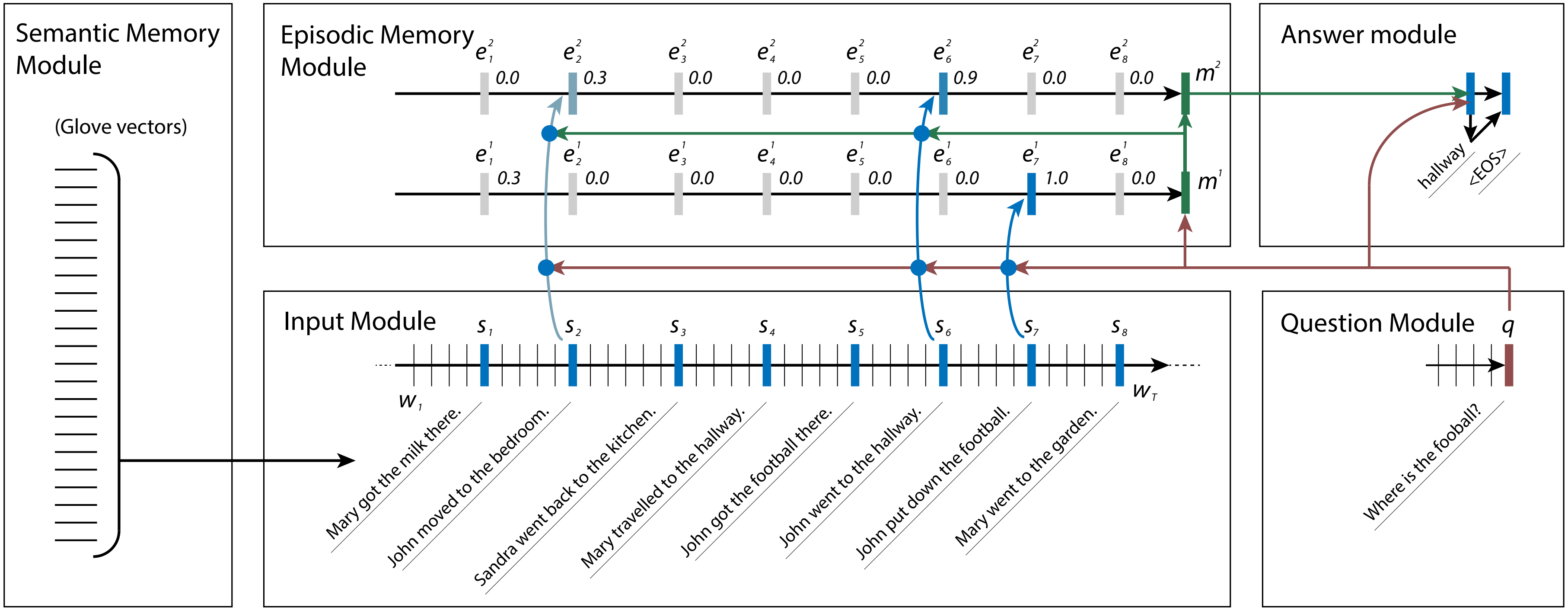
The Modules: Episodic Memory

- If summary is insufficient to answer the question, repeat sequence over input



The Modules: Answer

$$a_t = \texttt{GRU}([y_{t-1}, q], a_{t-1}), \qquad y_t = softmax(W^{(a)} a_t)$$



Related work

- Sequence to Sequence (Sutskever et al. 2014)
- Neural Turing Machines (Graves et al. 2014)
- Teaching Machines to Read and Comprehend (Hermann et al. 2015)
- Learning to Transduce with Unbounded Memory (Grefenstette 2015)
- Structured Memory for Neural Turing Machines (Wei Zhang 2015)

- Memory Networks (Weston et al. 2015)
- End to end memory networks (Sukhbaatar et al. 2015)
  →

Comparison to MemNets

Similarities:

- MemNets and DMNs have input, scoring, attention and response mechanisms

Differences:

- For input representations MemNets use bag of word, nonlinear or linear embeddings that explicitly encode position
- MemNets iteratively run functions for attention and response

- **DMNs show that neural sequence models can be used for input representation, attention and response mechanisms**
  → naturally captures position and temporality
- Enables broader range of applications

babl 1k, with gate supervision

Task	MemNN	DMN	Task	MemNN	DMN
1: Single Supporting Fact	100	100	11: Basic Coreference	100	99.9
2: Two Supporting Facts	100	98.2	12: Conjunction	100	100
3: Three Supporting facts	100	95.2	13: Compound Coreference	100	99.8
4: Two Argument Relations	100	100	14: Time Reasoning	99	100
5: Three Argument Relations	98	99.3	15: Basic Deduction	100	100
6: Yes/No Questions	100	100	16: Basic Induction	100	99.4
7: Counting	85	96.9	17: Positional Reasoning	65	59.6
8: Lists/Sets	91	96.5	18: Size Reasoning	95	95.3
9: Simple Negation	100	100	19: Path Finding	36	34.5
10: Indefinite Knowledge	98	97.5	20: Agent's Motivations	100	100
			Mean Accuracy (%)	93.3	**93.6**

Experiments: Sentiment Analysis

Stanford Sentiment Treebank

Test accuracies:

- MV-RNN and RNTN: Socher et al. (2013)
- DCNN: Kalchbrenner et al. (2014)
- PVec: Le & Mikolov. (2014)
- CNN-MC: Kim (2014)
- DRNN: Irsoy & Cardie (2015)
- CT-LSTM: Tai et al. (2015)

Task	Binary	Fine-grained
MV-RNN	82.9	44.4
RNTN	85.4	45.7
DCNN	86.8	48.5
PVec	87.8	48.7
CNN-MC	88.1	47.4
DRNN	86.6	49.8
CT-LSTM	88.0	51.0
DMN	**88.6**	**52.1**

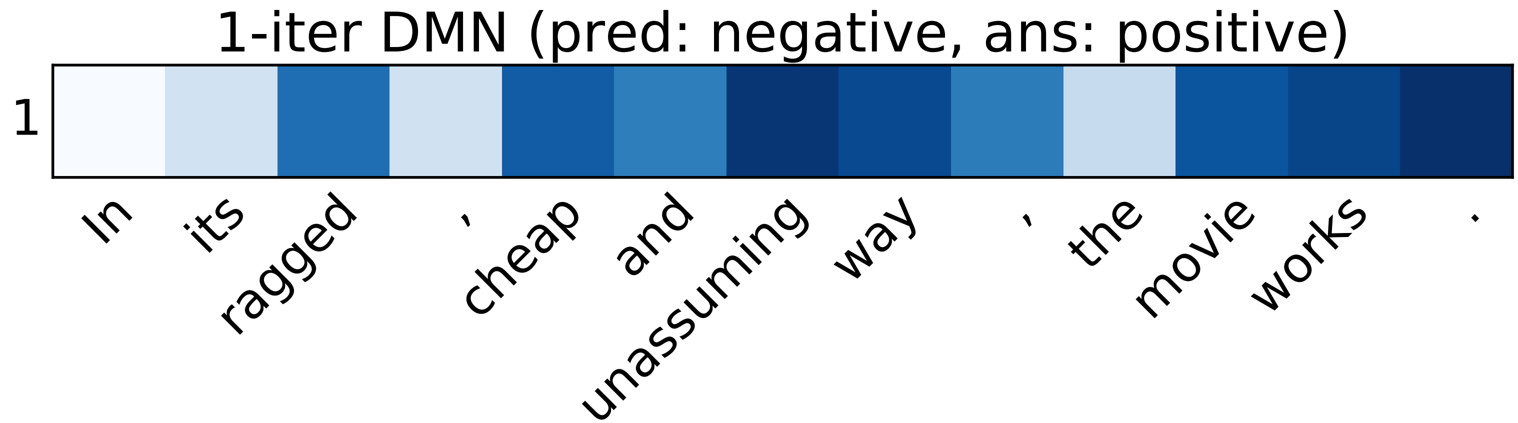
Analysis of Number of Episodes

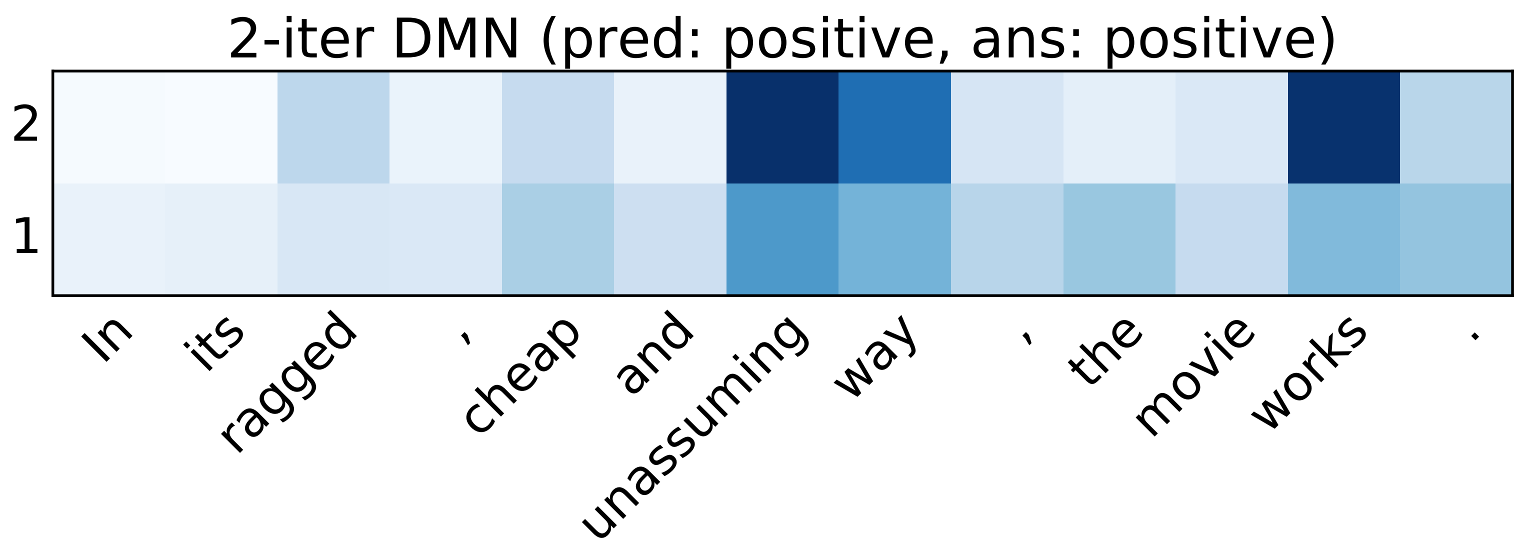
- How many attention + memory passes are needed in the episodic memory?

Max passes	task 3 three-facts	task 7 count	task 8 lists/sets	sentiment (fine grain)
0 pass	0	48.8	33.6	50.0
1 pass	0	48.8	54.0	51.5
2 pass	16.7	49.1	55.6	**52.1**
3 pass	64.7	83.4	83.4	50.1
5 pass	**95.2**	**96.9**	**96.5**	N/A

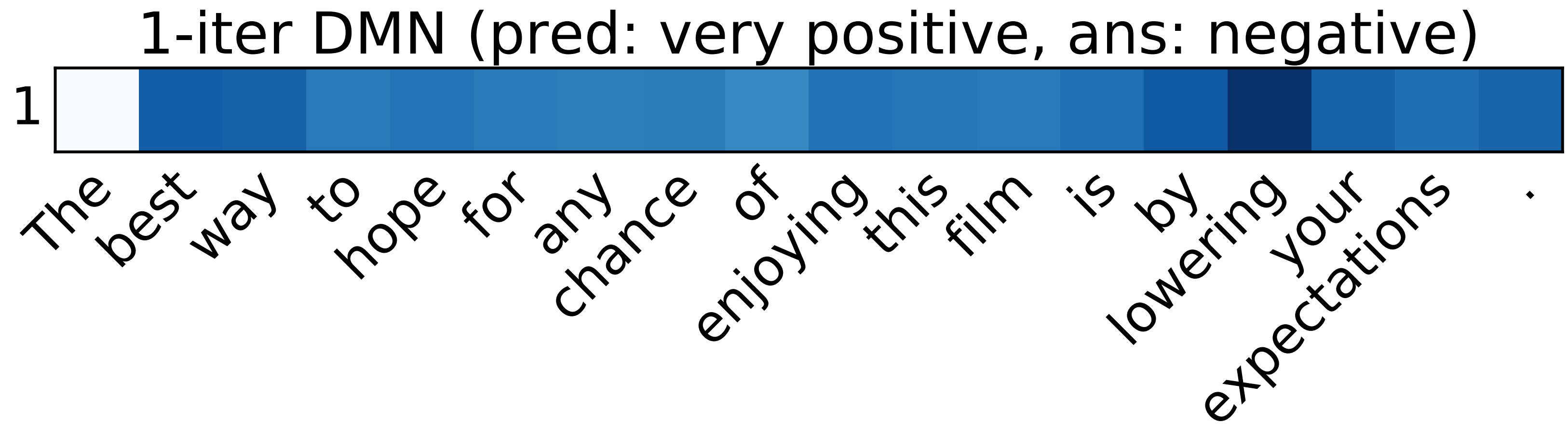
Analysis of Attention for Sentiment

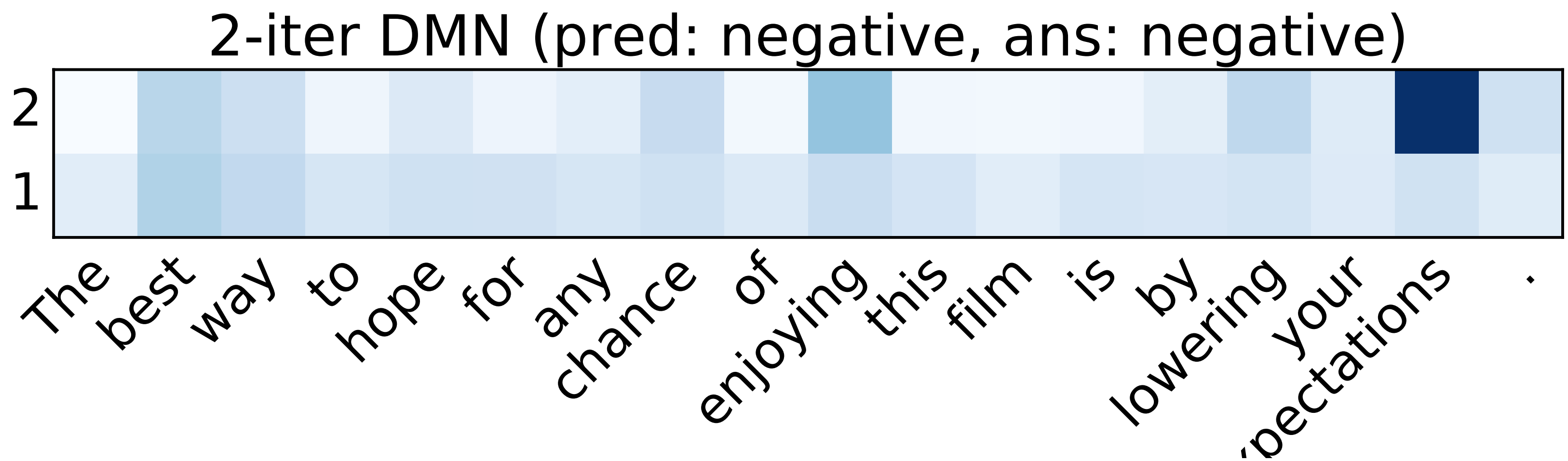
- Sharper attention when 2 passes are allowed.
- Examples that are wrong with just one pass





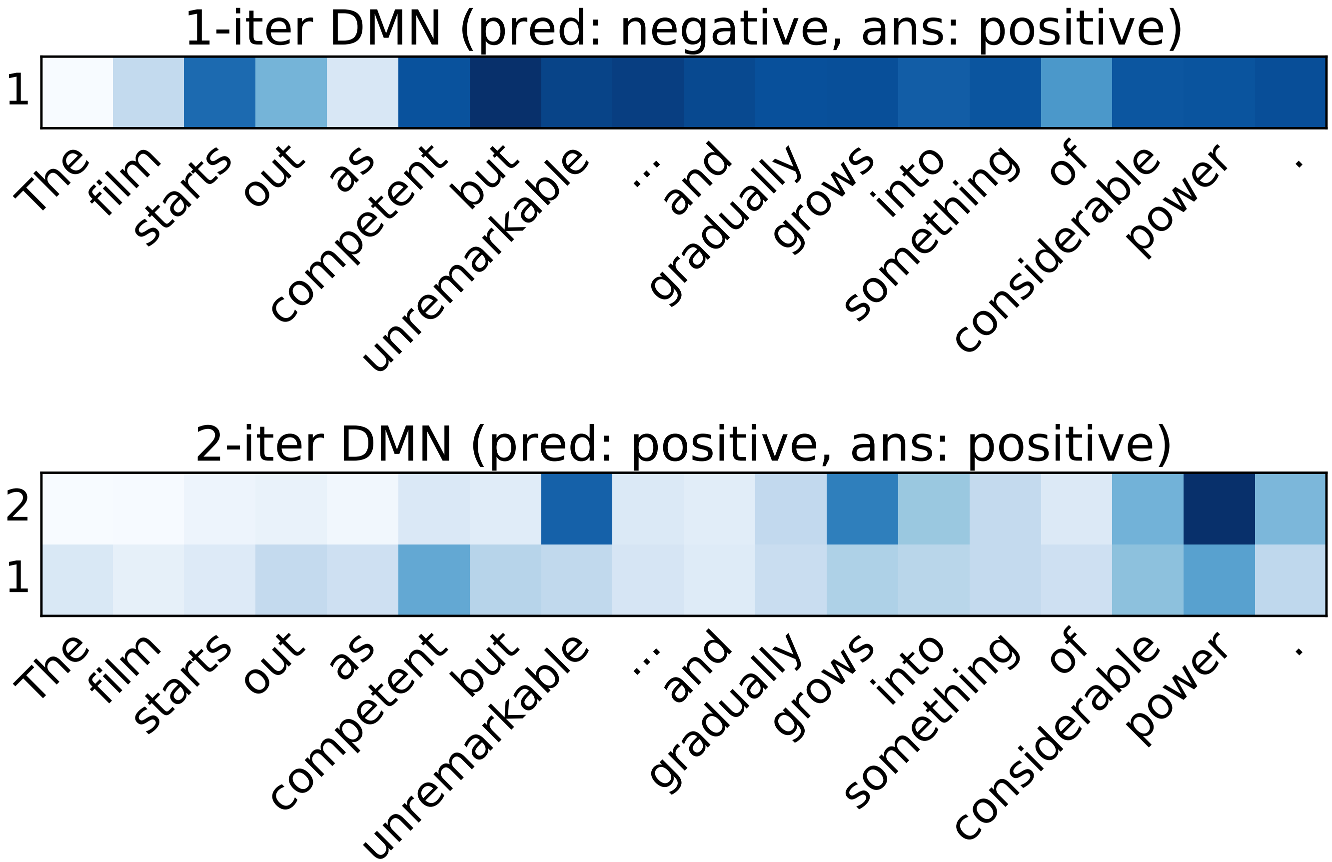
Analysis of Attention for Sentiment





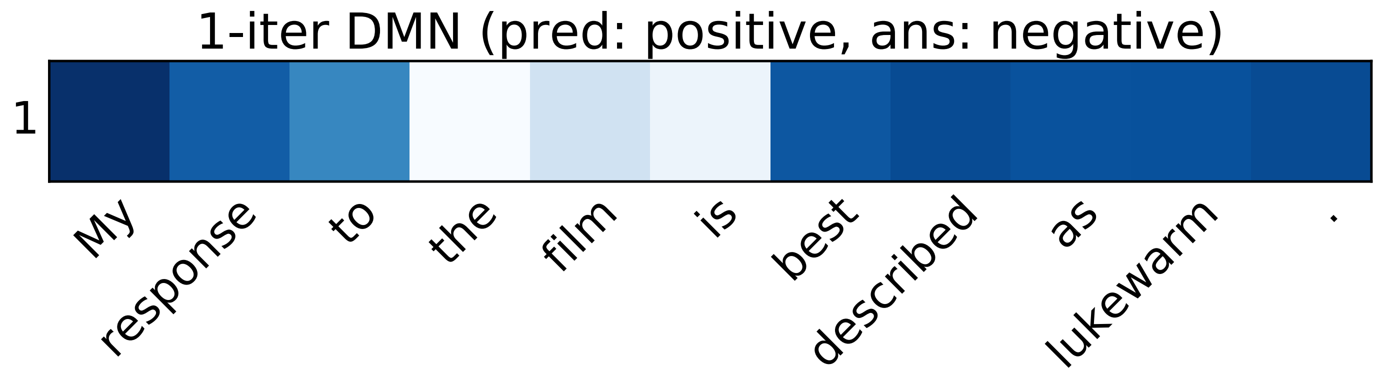
Analysis of Attention for Sentiment

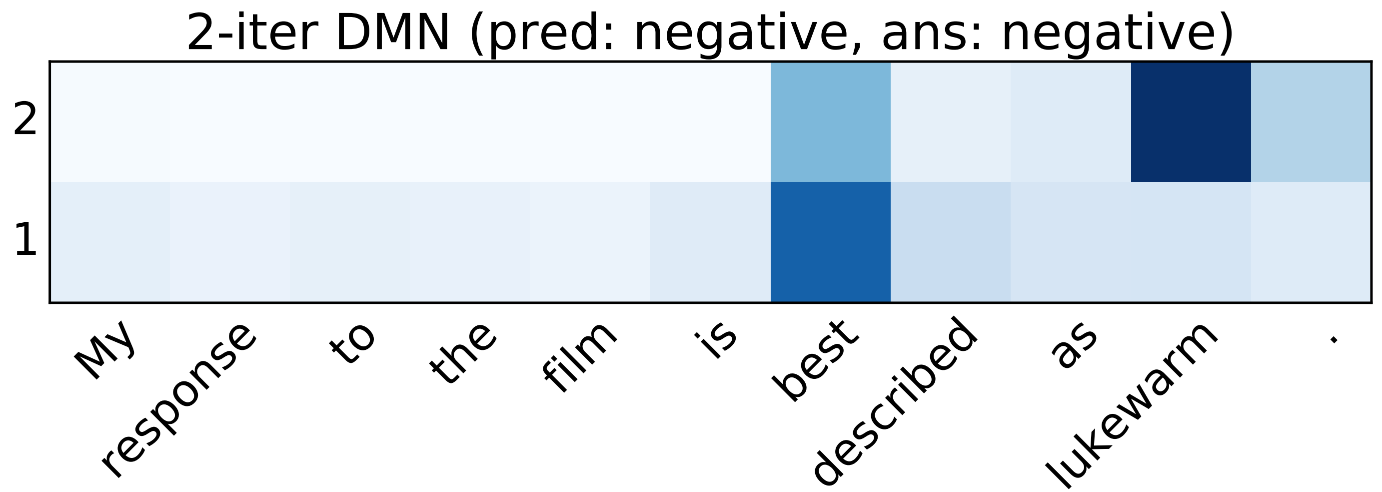
- Examples where full sentence context from first pass changes attention to words more relevant for final prediction



Analysis of Attention for Sentiment

- Examples where full sentence context from first pass changes attention to words more relevant for final prediction





Experiments: POS Tagging

- PTB WSJ, standard splits
- Episodic memory does not require multiple passes, single pass enough

Model	SVMTool	Sogaard	Suzuki et al.	Spoustova et al.	SCNN	DMN
Acc (%)	97.15	97.27	97.40	97.44	97.50	**97.56**

Modularization Allows for Different Inputs

Episodic Memory

Answer

Kitchen

Input Module

John moved to the garden.
John got the apple there.
John moved to the kitchen.
Sandra picked up the milk there.
John dropped the apple.
John moved to the office.

Question

Where is the apple?

Episodic Memory

Answer

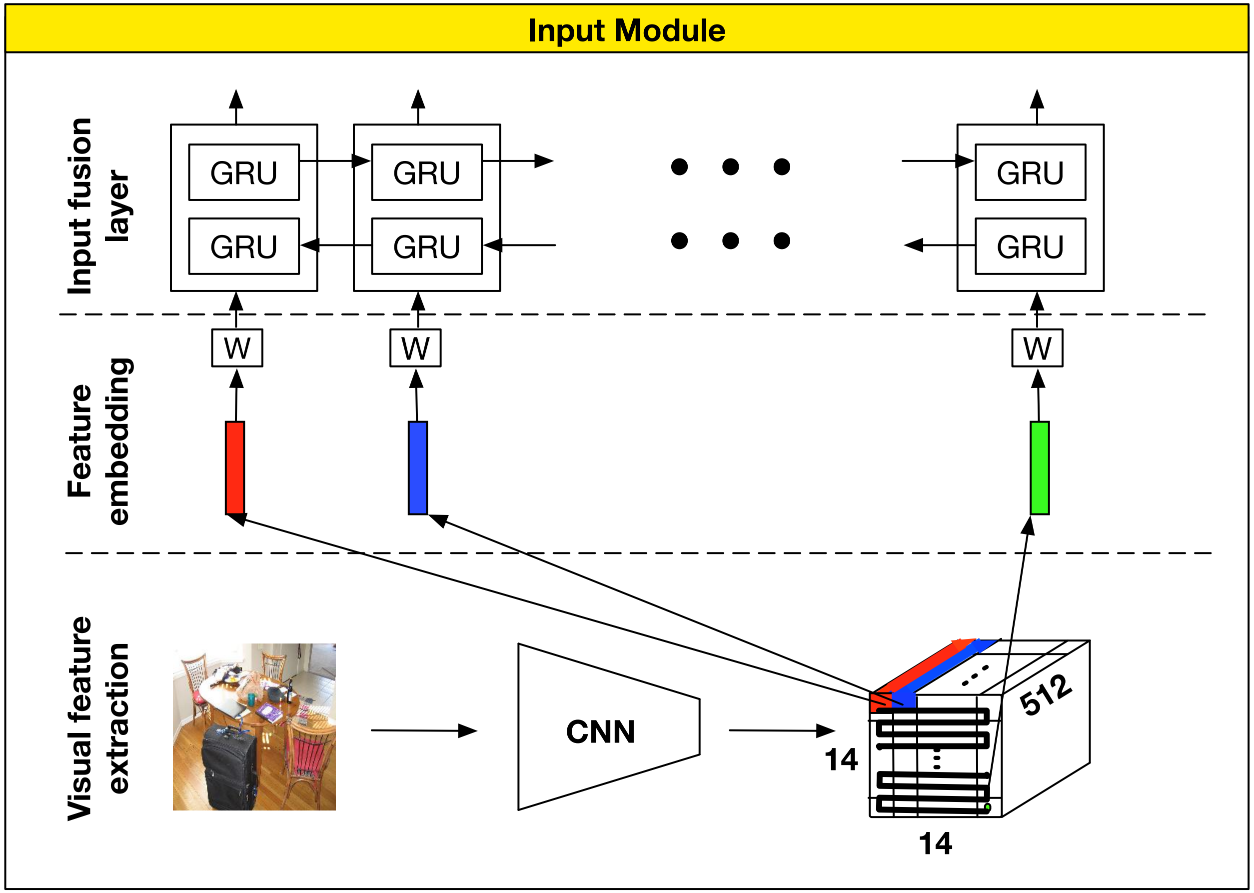
Palm

Input Module

Question

What kind of tree is in the background?

Input Module for Images



Dynamic Memory Networks for Visual and Textual Question Answering,
Caiming Xiong, Stephen Merity, Richard Socher

Accuracy: Visual Question Answering

VQA test-dev and test-standard:

- Antol et al. (2015)
- ACK Wu et al. (2015);
- iBOWIMG - Zhou et al. (2015);
- DPPnet - Noh et al. (2015); D-NMN - Andreas et al. (2016);
- SAN - Yang et al. (2015)

	test-dev				test-std
Method	All	Y/N	Other	Num	All
VQA					
Image	28.1	64.0	3.8	0.4	-
Question	48.1	75.7	27.1	36.7	-
Q+I	52.6	75.6	37.4	33.7	-
LSTM Q+I	53.7	78.9	36.4	35.2	54.1
ACK	55.7	79.2	40.1	36.1	56.0
iBOWIMG	55.7	76.5	42.6	35.0	55.9
DPPnet	57.2	80.7	41.7	37.2	57.4
D-NMN	57.9	80.5	43.1	37.4	58.0
SAN	58.7	79.3	46.1	36.6	58.9
DMN+	**60.3**	80.5	48.3	36.8	**60.4**

Attention Visualization

What is the main color on the bus ?

Answer: blue

What type of trees are in the background ?

Answer: pine

How many pink flags are there ?

Answer: 2

Is this in the wild ?

Answer: no

Attention Visualization

Which man is dressed more flamboyantly ? **Answer: right**

Who is on both photos ? **Answer: girl**

What time of day was this picture taken ? **Answer: night**

What is the boy holding ? **Answer: surfboard**

Attention Visualization

What is this sculpture made out of ?

Answer: metal

What color are the bananas ?

Answer: green

What is the pattern on the cat ' s fur on its tail ?

Answer: stripes

Did the player hit the ball ?

Answer: yes

What is the girl holding ?

tennis racket

What is the girl doing ?

playing tennis

Is the girl wearing a hat ?

yes

What is the girl wearing ?

shorts

What is the color of the ground ?

brown

What color is the ball ?

yellow

What color is her skirt ?

white

What did the girl just hit ?

tennis ball

Summary

- Basic blocks can be combined or learned with NAS
- Memory is useful. DMN accurately solves variety of tasks
- Next week: Most recent research and fun future outlook