Dynamic Memory Networks for Visual and Textual Question Answering

Caiming Xiong*, Stephen Merity*, Richard Socher

Presenters: **Zaccary Alperstein, Mohit Bajaj**

Motivation

Jim found a book.

He read few pages of the book and found it interesting.

He left for the university.

He took the book with him.

Where did Jim take the book?

Motivation

Jim found a book.

He read few pages of the book and found it interesting.

He left for the university.

He took it with him.

Where did Jim take the book?

Motivation

Jim found a book.

He read first few pages and found it interesting.

He left for the university.

He took it with him.

Where did Jim take the book?

Motivation

Jim found a book.

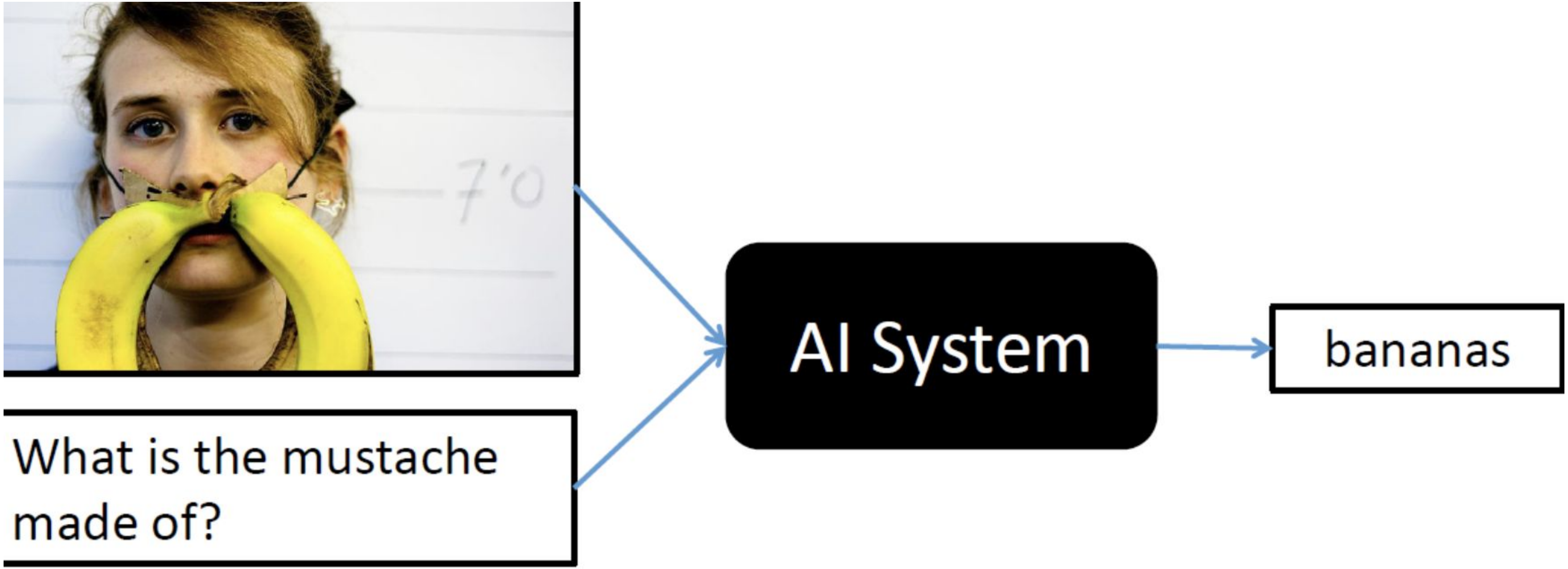
He read first few pages and found it interesting.

He left for the university.

He took it with him.

Where did Jim take the book? **University**

Visual QA



Motivation

- What if it was a complex article/ story and you are asked several questions?
  - Allowed to read once : Hard task!
  - You cannot memorize all at once
- Might need multiple glances over the facts to answer the question
  - Might need transitive reasoning
  - A lot easier this way
- Most of the other problems can be mapped to Q/A
  - Sentiment analysis : Given some input facts What's the sentiment?
  - POS tagging

Motivation

- What if it was a complex article/ story and you are asked several questions?
  - Allowed to read once : Hard task!
  - You cannot memorize all at once
- Might need multiple glances over the facts to answer the question
  - Might need transitive reasoning
  - A lot easier this way
- Most of the other problems can be mapped to Q/A
  - Sentiment analysis : Given some input facts What's the sentiment?
  - POS tagging

We want a general framework optimized for QA.

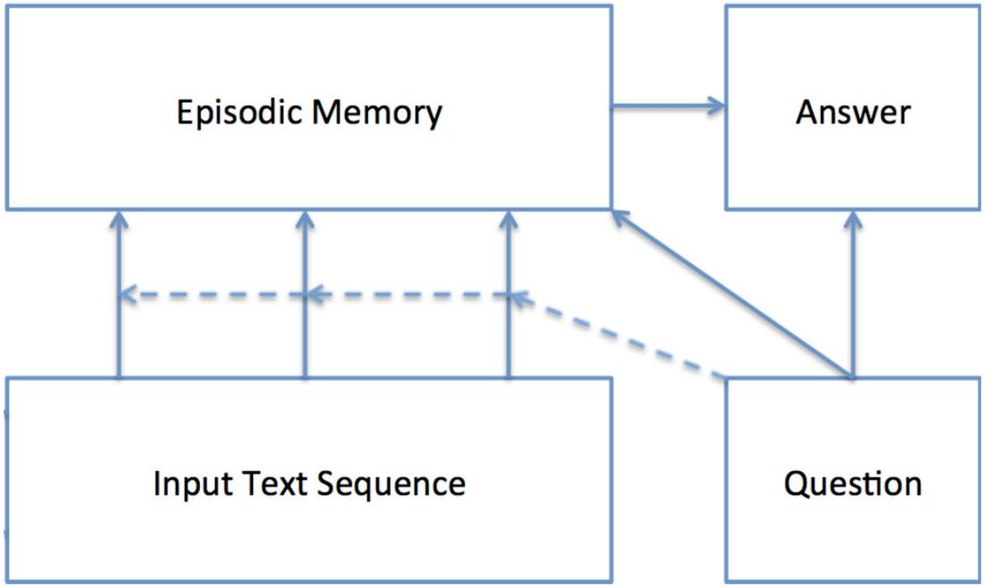
Related Work

- Memory Networks (Weston et al. 2015)
- Introduced memory networks for NLP QA
- Modules
  - **I** : (Input feature map) : Converts input to feature representation
  - **G** : (Generalization): Updates the old memory given new input
  - **O** : (Output feature map): Produces new output (in feature representation space) given the memories
  - **R** : (Response): Converts output to response seen by the world
- Hard attention
- Requires supervision at stages

Related Work

- End to end memory networks (Sukhbaatar et al. 2015)
    - Soft attention : Continuous
- Limitations:
    - Input sentences (facts) are processed independently
    - Needs extra features to capture positional information of sentences
    - Not applicable to variety of tasks
- Dynamic Memory Networks (Ask Me Anything: *Kumar et al.,2015)*
    - General architecture optimised for Q/A
    - Flow of information between facts
    - Generalization to other tasks
    - Achieved state-of-art results on tasks like POS tagging on some data-sets

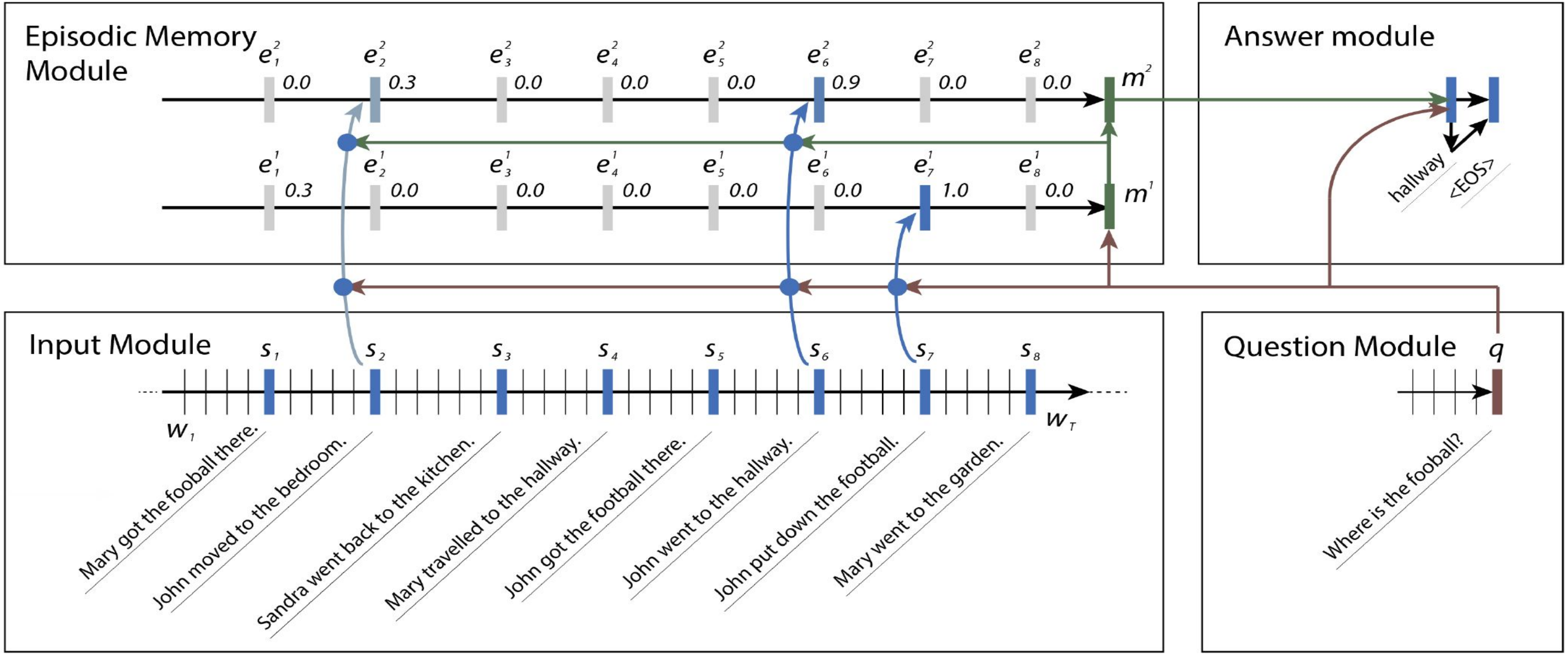
DMN: Model Overview



Modules

- Input Module
  - Encodes input facts through RNN (GRU)
  - One encoded representation for each sentence : ‘fact’

- Question Module
  - Encodes question through RNN similarly to input module


Episodic Memory Module
e_1^2 0.0 e_2^2 0.3 e_3^2 0.0 e_4^2 0.0 e_5^2 0.0 e_6^2 0.9 e_7^2 0.0 e_8^2 0.0
m^2
e_1^1 0.3 e_2^1 0.0 e_3^1 0.0 e_4^1 0.0 e_5^1 0.0 e_6^1 0.0 e_7^1 1.0 e_8^1 0.0
m^1
Answer module
hallway
<EOS>
Input Module
s_1 s_2 s_3 s_4 s_5 s_6 s_7 s_8
w_1
w_T
Mary got the fooball there.
John moved to the bedroom.
Sandra went back to the kitchen.
Mary travelled to the hallway.
John got the football there.
John went to the hallway.
John put down the football.
Mary went to the garden.
Question Module
q
Where is the fooball?

Episodic Memory Module

- Memory is updated after each episode
- Attention mechanism
  - Triggered by the question to find relevant facts conditioned on previous memory
  - 2 layer MLP to compute attention from the similarity vector

$$z(c,m,q) = \left[c, m, q, c \circ q, c \circ m, |c-q|, |c-m|, c^T W^{(b)} q, c^T W^{(b)} m\right] \tag{5}$$

$$G(c,m,q) =$$

$$\sigma\left(W^{(2)} \tanh\left(W^{(1)} z(c,m,q) + b^{(1)}\right) + b^{(2)}\right).$$

Episodic Memory Module

- Attention + RNN
  - GRU to aggregate the attention over facts

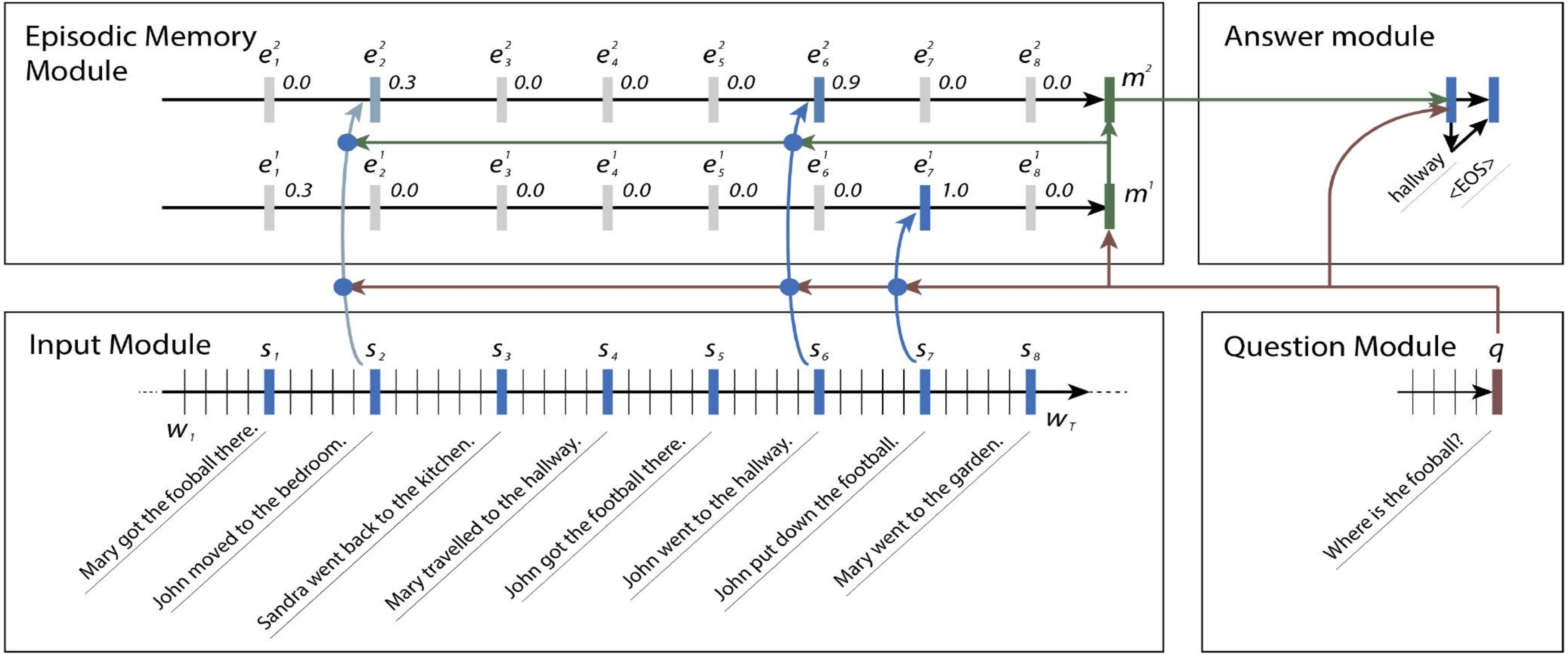
$$
\begin{aligned}
h_t^i &= g_t^i GRU(c_t, h_{t-1}^i) + (1 - g_t^i) h_{t-1}^i \\
e^i &= h_{T_C}^i
\end{aligned}
$$

Episodic memory

- Multiple passes over input : Why?
  - Transitive reasoning
  - Prevents over-burdening the attention

- $m^i = GRU(e^i, m^{i-1})$, $m^0 = q$

- After T_m passes, m^{Tm} is used to decode the answer



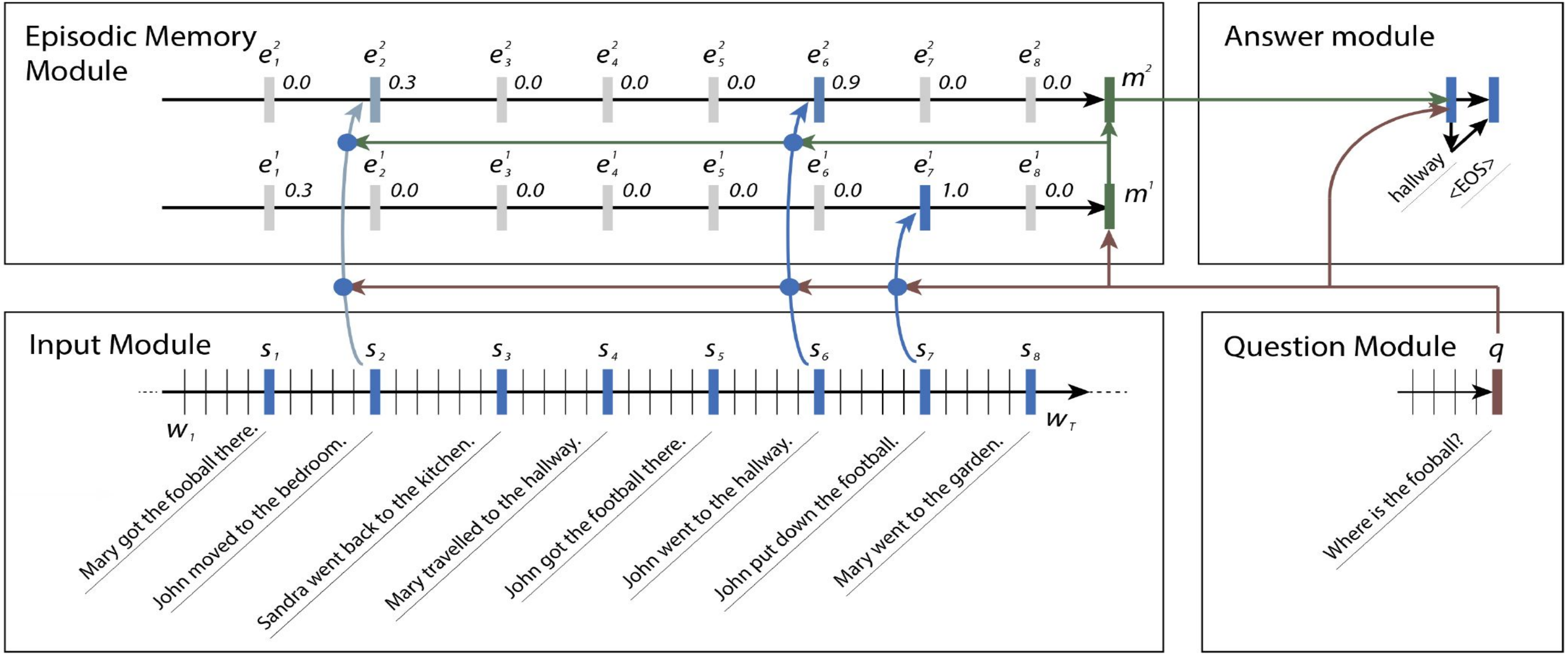
Episodic Memory Module
e_1^2 0.0 e_2^2 0.3 e_3^2 0.0 e_4^2 0.0 e_5^2 0.0 e_6^2 0.9 e_7^2 0.0 e_8^2 0.0
m^2
e_1^1 0.3 e_2^1 0.0 e_3^1 0.0 e_4^1 0.0 e_5^1 0.0 e_6^1 0.0 e_7^1 1.0 e_8^1 0.0
m^1
Answer module
hallway
<EOS>
Input Module
s_1 s_2 s_3 s_4 s_5 s_6 s_7 s_8
w_1
w_T
Mary got the fooball there.
John moved to the bedroom.
Sandra went back to the kitchen.
Mary travelled to the hallway.
John got the football there.
John went to the hallway.
John put down the football.
Mary went to the garden.
Question Module
q
Where is the fooball?

Answer module

- Decodes the answer given final memory state, previously generated output and the question

$$
\begin{aligned}
a_0 &= m^{T_M} \\
y_t &= \operatorname{softmax}(W^{(a)} a_t) \\
a_t &= GRU([y_{t-1}, q], a_{t-1}),
\end{aligned}
$$

- Loss
  - Cross-entropy of answer sequence
  - Cross entropy of attention gates, if available in dataset



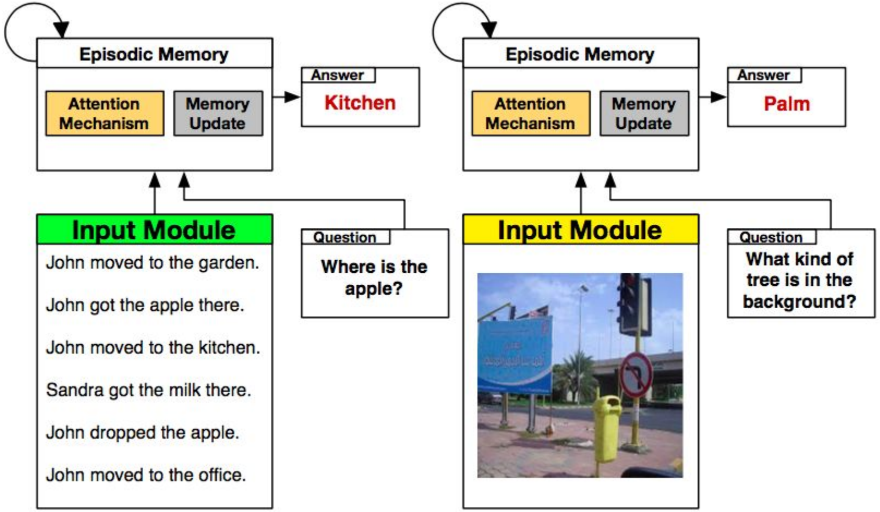
Episodic Memory Module
Answer module
Input Module
Question Module
e^2_1 0.0
e^2_2 0.3
e^2_3 0.0
e^2_4 0.0
e^2_5 0.0
e^2_6 0.9
e^2_7 0.0
e^2_8 0.0
m^2
e^1_1 0.3
e^1_2 0.0
e^1_3 0.0
e^1_4 0.0
e^1_5 0.0
e^1_6 0.0
e^1_7 1.0
e^1_8 0.0
m^1
hallway
<EOS>
s_1
s_2
s_3
s_4
s_5
s_6
s_7
s_8
w_1
w_T
q
Mary got the fooball there.
John moved to the bedroom.
Sandra went back to the kitchen.
Mary travelled to the hallway.
John got the football there.
John went to the hallway.
John put down the football.
Mary went to the garden.
Where is the fooball?

DMN/ DMN+ Four Main Ingredients

1. Input Module
- Process the raw text or image into hidden or 'fact' vectors
- Includes novel **Input fusion layer**

2. Question Module
- Encodes question
- Usually just final hidden output of GRU

3. Episodic Memory Module
- Chooses important parts of inputs to **pay attention** to, **outputting a memory vector**
- **Multiple passes** along input encodings



(a) Text Question-Answering (b) Visual Question-Answering

4. Answer Module
- Receives question representation, and 'memory' output from episodic memory, to output:
- Can output distribution over single class, or sequential

DMN/ DMN+ Input Module

DMN

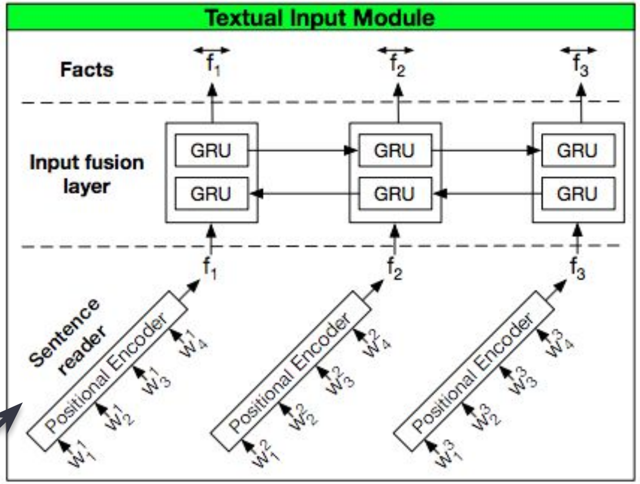
- When there is a single sentence, use hidden representation per word
- When there are multiple sentences, concatenate words in each sentence, output hidden vector for each sentence
- Would have to do a lot of awkward padding here

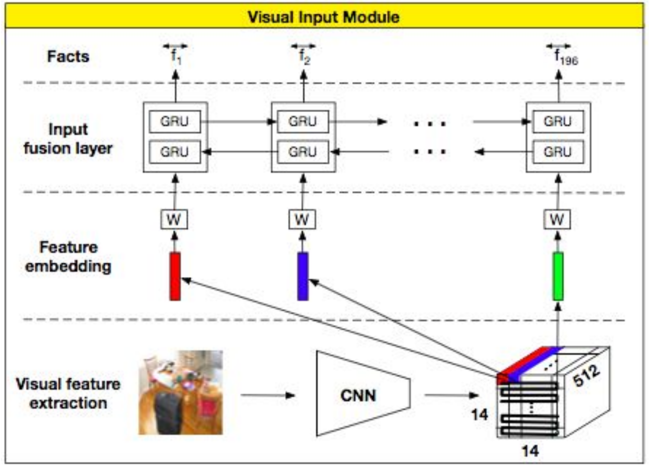
DMN+

- **Positional Encoder** used to process words in sentence:

$$f_i = \sum_M^{j=1} l_j \circ w_j^i$$

- Where $\mathbf{w_i}$ is a word representation and $\mathbf{l_j}$ is an unparametrized positional embedding
- Weighted sum used to add word representations
- Images simply use VGG-19 vectors



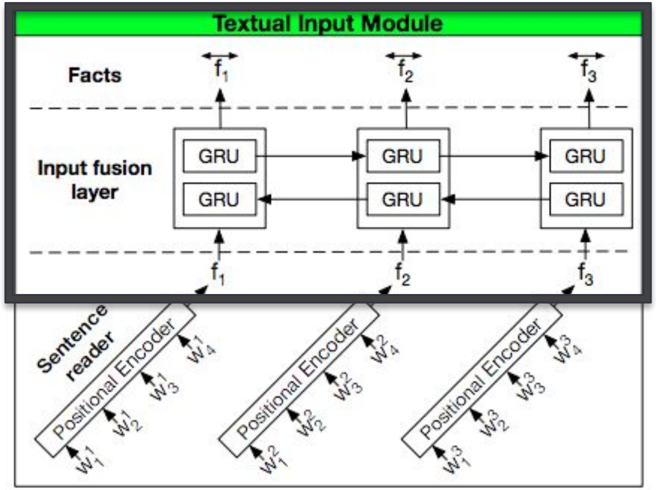


DMN+ Input Module

Bidirectional- GRU encodes positional Information in **input fusion layer**
- Gradient pathway reduced in comparison to single GRU

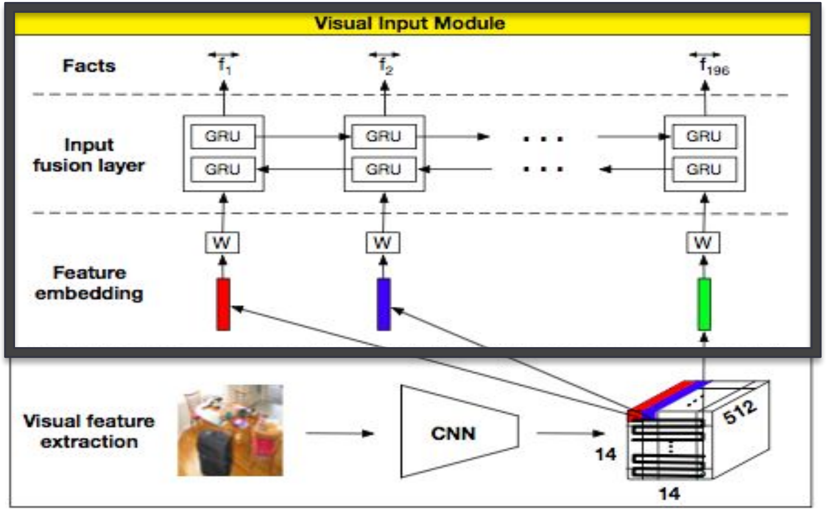
Text

- Encodes global information context into **'fact'** vectors



Images

- Encodes local regions of images into global image representation
- Scale input images to equal size



DMN+ Episodic Memory Module: Attention

- Takes ‘fact’ vectors from ‘input fusion layer’ as input
- Builds attention vectors
- Query with minimal parameters (DMN was similar, but had a few extra parameters)

Attention Mechanisms: Ingredients

1. **Query vector**
   - question vector representation and previous **memory** vector (created recursively)
2. **Set of value vectors to reference**
   - Here these are the ‘fact’ vectors created by the input fusion module
3. **Similarity Function**
   - Throw together a bunch of **similarity measures** between facts, query, and memory, then push through **fully connected layer**, and **normalize**

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

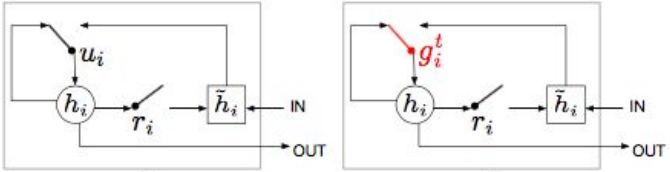
Episodic Memory Module: Attention and Update

DMN+

1. **GRU based attention mechanism**

- The weight vector is then used as a scalar ‘gating’ mechanism

$$h_i = g_i^t \circ \tilde{h}_i + (1 - g_i^t) \circ h_{i-1}$$



(a) (b)

- Scalar attention **g_i for each fact i**
- Forms ‘attention mechanism’ output after GRU has run over ‘fact’ vectors
- Alternative update just convex sum of fact vectors (vanilla attention)

2. **GRU based memory update**

- Output from GRU attention mechanism used to update the **memory state** with another GRU

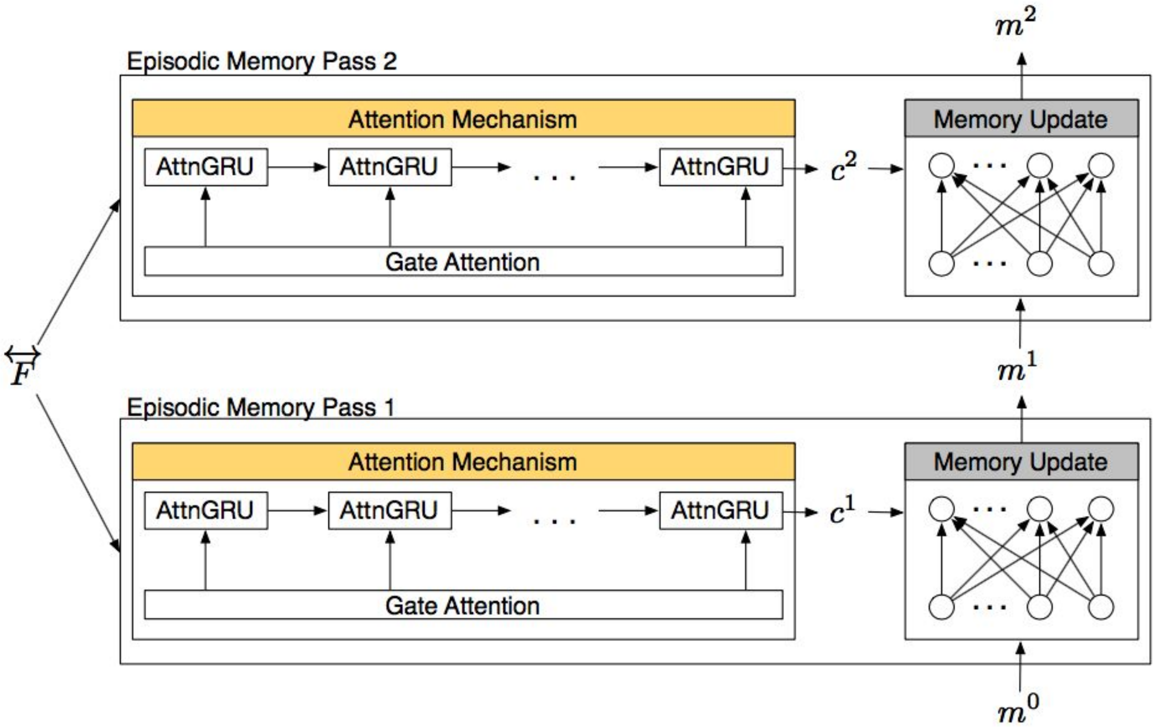
Steps 1 and 2 are repeated for multiple episodes (3 here) taking multiple glimpses at the input

Con:
-This is just a scalar! Decreases representational power

- Gating meant to **send gradients to zero**, this mechanism doesn’t do that
- **A lot** of facts means update is **minimal, small** number of facts leads to **large updates**

DMN/ DMN+ Episodic Memory Module: Attention

DMN+



GRU if tied, otherwise

$$m^t = ReLU\left(W^t[m^{t-1}; c^t; q] + b\right)$$

Con:
- makes it less 'dynamic'
As it cannot take arbitrary number of passes on facts

DMN+ Final Output: Answer Module

- Just like the old DMN

Where **a** is the last memory, and a GRU may be used in the case of a sequence output

$$
\begin{aligned}
y_t &= \mathrm{softmax}(W^{(a)} a_t) \\
a_t &= GRU([y_{t-1}, q], a_{t-1}),
\end{aligned}
$$

DMN/ DMN+ Results

Con:

Likely Irreproducible

> On some tasks, the accuracy was not stable across multiple runs. This was particularly problematic on QA3, QA17, and QA18. To solve this, we repeated training 10 times using random initializations and evaluated the model that achieved the lowest validation set loss.

- **From this statement it is no longer clear that anything they did in their network design was actually useful**
- **Their model was unstable, and they didn't report standard deviations or averages**
- **Could have just gotten lucky, expect <u>attractors</u> in good architectures**

DMN/DMN+ Results: Data sets

bAbI-10k

- Synthetic dataset
- 20 different questions
- Composed of a set of **facts** and at least one **question**

Example questions

Task 1: Single Supporting Fact
Mary went to the bathroom.
John moved to the hallway.
Mary travelled to the office.
Where is Mary? A:office

Task 2: Two Supporting Facts
John is in the playground.
John picked up the football.
Bob went to the kitchen.
Where is the football? A:playground

Task 3: Three Supporting Facts
John picked up the apple.
John went to the office.
John went to the kitchen.
John dropped the apple.
Where was the apple before the kitchen? A:office

Task 4: Two Argument Relations
The office is north of the bedroom.
The bedroom is north of the bathroom.
The kitchen is west of the garden.
What is north of the bedroom? A: office
What is the bedroom north of? A: bathroom

Task 5: Three Argument Relations
Mary gave the cake to Fred.
Fred gave the cake to Bill.
Jeff was given the milk by Bill.
Who gave the cake to Fred? A: Mary
Who did Fred give the cake to? A: Bill

Task 6: Yes/No Questions
John moved to the playground.
Daniel went to the bathroom.
John went back to the hallway.
Is John in the playground? A:no
Is Daniel in the bathroom? A:yes

DMN/DMN+ Results: Data sets

Task 7: Counting
Daniel picked up the football.
Daniel dropped the football.
Daniel got the milk.
Daniel took the apple.
How many objects is Daniel holding? A: two

Task 8: Lists/Sets
Daniel picks up the football.
Daniel drops the newspaper.
Daniel picks up the milk.
John took the apple.
What is Daniel holding? milk, football

Task 9: Simple Negation
Sandra travelled to the office.
Fred is no longer in the office.
Is Fred in the office? A:no
Is Sandra in the office? A:yes

Task 10: Indefinite Knowledge
John is either in the classroom or the playground.
Sandra is in the garden.
Is John in the classroom? A:maybe
Is John in the office? A:no

Task 11: Basic Coreference
Daniel was in the kitchen.
Then he went to the studio.
Sandra was in the office.
Where is Daniel? A:studio

Task 12: Conjunction
Mary and Jeff went to the kitchen.
Then Jeff went to the park.
Where is Mary? A: kitchen
Where is Jeff? A: park

Task 13: Compound Coreference
Daniel and Sandra journeyed to the office.
Then they went to the garden.
Sandra and John travelled to the kitchen.
After that they moved to the hallway.
Where is Daniel? A: garden

Task 14: Time Reasoning
In the afternoon Julie went to the park.
Yesterday Julie was at school.
Julie went to the cinema this evening.
Where did Julie go after the park? A:cinema
Where was Julie before the park? A:school

DMN/DMN+ Results: Data sets

Task 15: Basic Deduction
Sheep are afraid of wolves.
Cats are afraid of dogs.
Mice are afraid of cats.
Gertrude is a sheep.
What is Gertrude afraid of? A:wolves

Task 16: Basic Induction
Lily is a swan.
Lily is white.
Bernhard is green.
Greg is a swan.
What color is Greg? A:white

Task 17: Positional Reasoning
The triangle is to the right of the blue square.
The red square is on top of the blue square.
The red sphere is to the right of the blue square.
Is the red sphere to the right of the blue square? A:yes
Is the red square to the left of the triangle? A:yes

Task 18: Size Reasoning
The football fits in the suitcase.
The suitcase fits in the cupboard.
The box is smaller than the football.
Will the box fit in the suitcase? A:yes
Will the cupboard fit in the box? A:no

Task 19: Path Finding
The kitchen is north of the hallway.
The bathroom is west of the bedroom.
The den is east of the hallway.
The office is south of the bedroom.
How do you go from den to kitchen? A: west, north
How do you go from office to bathroom? A: north, west

Task 20: Agent's Motivations
John is hungry.
John goes to the kitchen.
John grabbed the apple there.
Daniel is hungry.
Where does Daniel go? A:kitchen
Why did John go to the kitchen? A:hungry

DMN/DMN+ Results: VQA Dataset / DAQUAR-ALL

VQA:MS-COCO

- Classic MS-COCO based VQA dataset
- 123,287 training/validation images and 81,434 test images
- Each image has several related questions, each being answered by multiple people
- 248,349 training questions, 121512 validation questions, 244,302 testing questions (test set split in test-standard and test-challenge)
- Evaluation on both test-standard and test-challenged implemented via submission system, test standard may only be evaluated 5 times and test-challenge at the end of the competition

DAQUAR-ALL

- Dataset for question answering on real world images
- Consists of 795 training images and 654 test images
- 6795 training questions, 5673 test
- Multiple word answers excluded

DMN/DMN+ Results: Ablation and comparison

Yellow: Regular/GRU attention

Green: GRU/ Dense memory update

ODMN: Original DMN

DMN2: Replaces input module with fusion layer
- Makes for most of the increase in accuracy
- The questions are highly positionally dependent on the facts input

DMN3: Adds GRU attention mechanism to DMN2

DMN+: Uses untied weights for memory update on DMN3

CON: ODMN did three times better on bAbi1k, missing comparison!!

Model	ODMN	DMN2	DMN3	DMN+
Input module	GRU	Fusion	Fusion	Fusion
Attention	$\sum g_i f_i$	$\sum g_i f_i$	AttnGRU	AttnGRU
Mem update	GRU	GRU	GRU	ReLU
Mem Weights	Tied	Tied	Tied	Untied
bAbI English 10k dataset				
QA2	36.0	0.1	0.7	0.3
QA3	42.2	19.0	9.2	1.1
QA5	0.1	0.5	0.8	0.5
QA6	35.7	0.0	0.6	0.0
QA7	8.0	2.5	1.6	2.4
QA8	1.6	0.1	0.2	0.0
QA9	3.3	0.0	0.0	0.0
QA10	0.6	0.0	0.2	0.0
QA14	3.6	0.7	0.0	0.2
QA16	55.1	45.7	47.9	45.3
QA17	39.6	5.9	5.0	4.2
QA18	9.3	3.8	0.1	2.1
QA20	1.9	0.0	0.0	0.0
Mean error	11.8	3.9	3.3	2.8
DAQUAR-ALL visual dataset				
Accuracy	27.54	28.43	28.62	28.79

Note: 0 error on questions not shown

DMN/DMN+ Results: SOTA comparison

E2E: End-to-End Memory Network

- Similarly builds memory representation

NR: Neural Reasoner

- More basic multi-level RNN which produces multiple beliefs based on facts and questions with higher layers, that pool outputs

Note: questions where both models got 0 error not shown

Task	DMN+	E2E	NR
2: 2 supporting facts	0.3	0.3	-
3: 3 supporting facts	1.1	2.1	-
5: 3 argument relations	0.5	0.8	-
6: yes/no questions	0.0	0.1	-
7: counting	2.4	2.0	-
8: lists/sets	0.0	0.9	-
9: simple negation	0.0	0.3	-
11: basic coreference	0.0	0.1	-
14: time reasoning	0.2	0.1	-
16: basic induction	45.3	51.8	-
17: positional reasoning	4.2	18.6	0.9
18: size reasoning	2.1	5.3	-
19: path finding	0.0	2.3	1.6
Mean error (%)	2.8	4.2	-
Failed tasks (err >5%)	1	3	-

DMN/DMN+ Results: VQA- MS COCO

-VQA dataset each question is answered by multiple people and answers may not be the same

- For each predicted answer the accuracy metric assigns **100% if at least 3 people provide the exact same answer**

Models in two classes:

1. Those the perform reasoning over multiple images patches
   - **SAN and DMN**

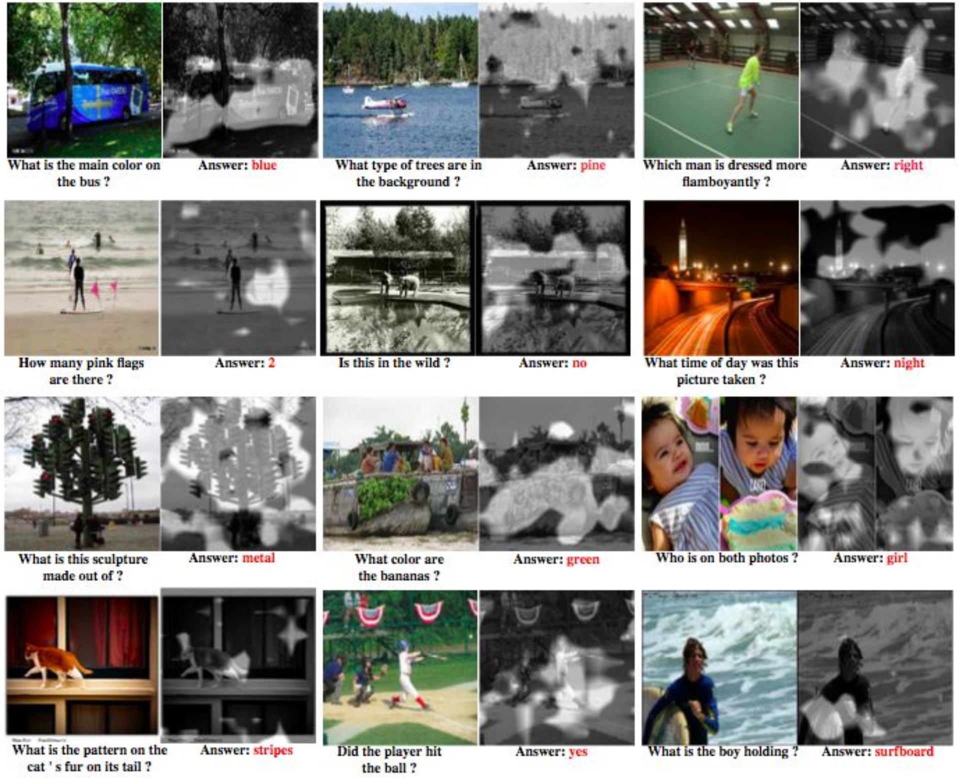
1. Those that utilize a full connected image feature for classification
   - **Everything else**

	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**

Reason that they did worse in number based questions likely because of the image patches used, which is sometimes beneficial.

DMN/DMN+ Results: Qualitative Results on VQA

Attention scalar seems to be doing it’s job



Possible extensions

- More Regularization
  - **RNN** probably over-fitting the most
  - **Dropout inputs and outputs, variational recurrent dropout, zoneout**
  - Currently only dropout vgg inputs, and final vector
- Parametrizing sentence representations
  - This should work better than **ad hoc** sentence representations, just need to regularize it a lot
- Gated convolutional encoding
  - Less parameters, easier to train then RNNs
  - Instead of GRU in episodic memory module
- Attention vector output instead of scalar, with sigmoid activation
  - Saying we should simultaneously pay attention to multiple things (I think that's ok)

Thank you!

DMN/ DMN+ Input Module

DMN

- When there is a single sentence, use hidden representation per word

- When there are multiple sentences, concatenate words in each sentence, output hidden vector for each sentence

  - would have to do a lot of awkward padding here

DMN+

- **Positional Encoder** used to process words in sentence:

$$f_i = \sum_M^{j=1} l_j \circ w_j^i$$

- Where **w_i** is a word representation and **l_j** is an unparametrized positional embedding

- weighted sum used to add word representations

- Images simply use VGG-19 vectors