Ask Me Anything:
Dynamic Memory Networks for Natural Language Processing

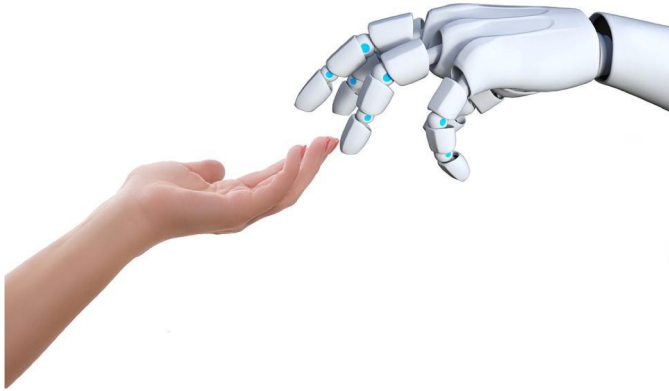
Ankit Kumar, Peter Ondruska, Mohit Iyyer, James Bradbury, Ishaan Gulrajani, Victor Zhong, Romain Paulus, Richard Socher
firstname@metamind.io, MetaMind, Palo Alto, CA USA

Proceedings of The 33rd International Conference on Machine Learning, PMLR 48:1378-1387, 2016.

Dynamic Neural Networks for Question Answering

Presented by Lisa Zhao

Outline

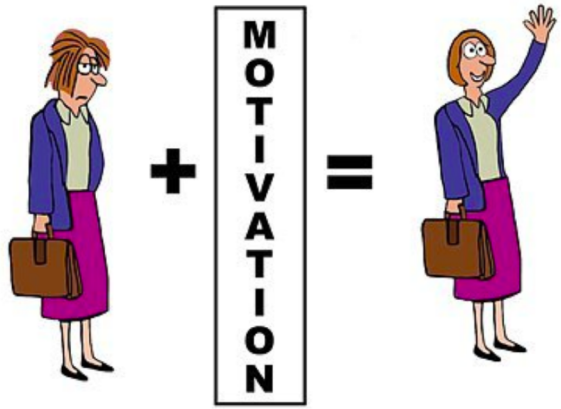


- Motivation & Overview

- Dynamic Memory Network [DMN] Explanation
- DMN Experimentations & Results

- Related Work & Comparison
- Conclusion & Discussion

Outline



- Motivation & Overview

- Dynamic Memory Network [DMN] Explanation
- DMN Experimentations & Results

- Related Work & Comparison
- Conclusion & Discussion

Motivation

An obstacle for general NLP is that different tasks (such as text classification, sequence tagging and text generation) require different sequential architectures. For example:

NLP task	State of the art model
Sentiment Analysis (IMDb)	XLNet (Yang et al., 2019)
POS Tagging (WSJ of Penn Treebank)	Meta BiLSTM (Bohnet et al., 2018)
Machine Translation (English-German of WMT)	Transformer Big + BT (Edunov et al., 2018)

*Information collected from http://nlpprogress.com/ and is up-to-date

Motivation

One way to deal with this problem is to view these different tasks as question-answering problems. For example:

Part-of-speech tagging (POS)

I: It started boring, but then it got interesting.
Q: POS tags?
A: PRP VBD JJ , CC RB PRP VBD JJ .

Question Answering

I: Jane went to the hallway.
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

Overview

A natural next step given the motivations would be to create

A single joint model for general Q&A tasks

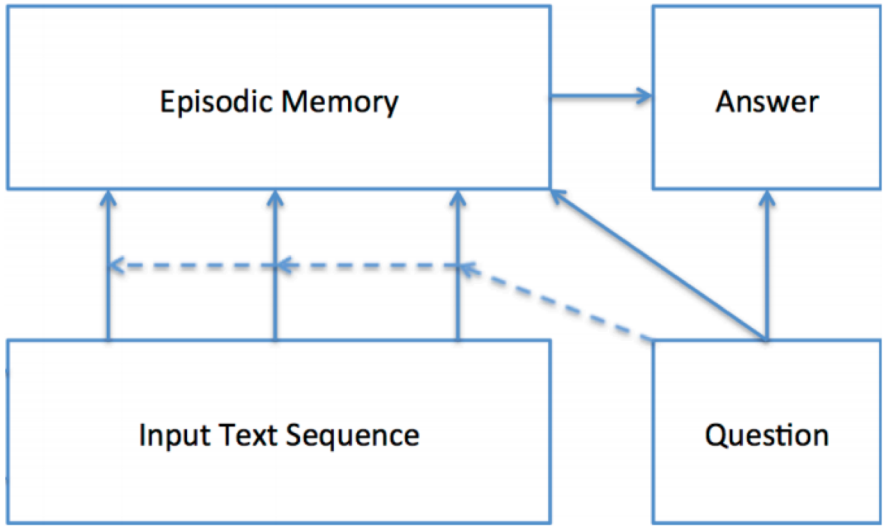
The paper “Ask Me Anything: Dynamic Memory Networks for Natural Language Processing” introduces the dynamic memory network (**DMN**), a neural network architecture which processes input sequences and questions, forms episodic memories, and generates relevant answers. Simply put,

DMN is a new, modularised architecture for Q&A

Overview

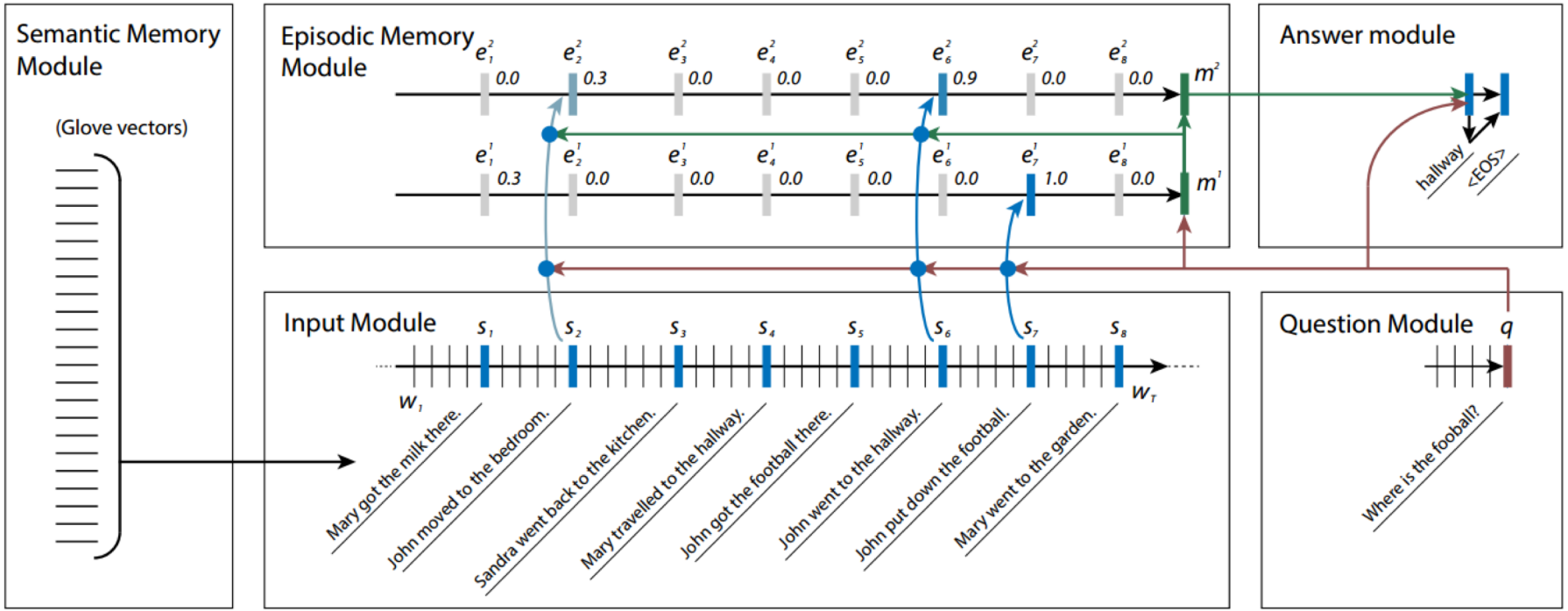
Four modules for Dynamic Memory Networks (DMN):

1) Input module
2) Question module
3) Episodic memory module
4) Answer module



(Kumar et al., 2016)

Dynamic Memory Network



(Kumar et al., 2016)

Let’s walk through each of the modules

- Motivation & Overview

- Dynamic Memory Network [DMN] Explanation
- DMN Experimentations & Results

- Related Work & Comparison
- Conclusion & Discussion

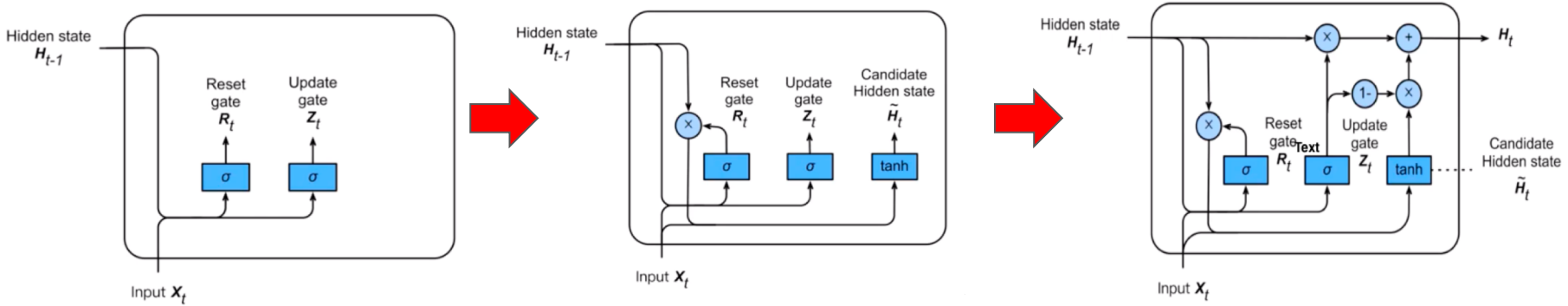
Brief Overview of Gated Recurrent Unit (GRU)

GRU is heavily used in the Dynamic Memory Network. It aims to solve the **vanishing gradient problem.**

$$\boldsymbol{R}_t = \sigma(\boldsymbol{X}_t \boldsymbol{W}_{xr} + \boldsymbol{H}_{t-1} \boldsymbol{W}_{hr} + \boldsymbol{b}_r),$$
$$\boldsymbol{Z}_t = \sigma(\boldsymbol{X}_t \boldsymbol{W}_{xz} + \boldsymbol{H}_{t-1} \boldsymbol{W}_{hz} + \boldsymbol{b}_z)$$

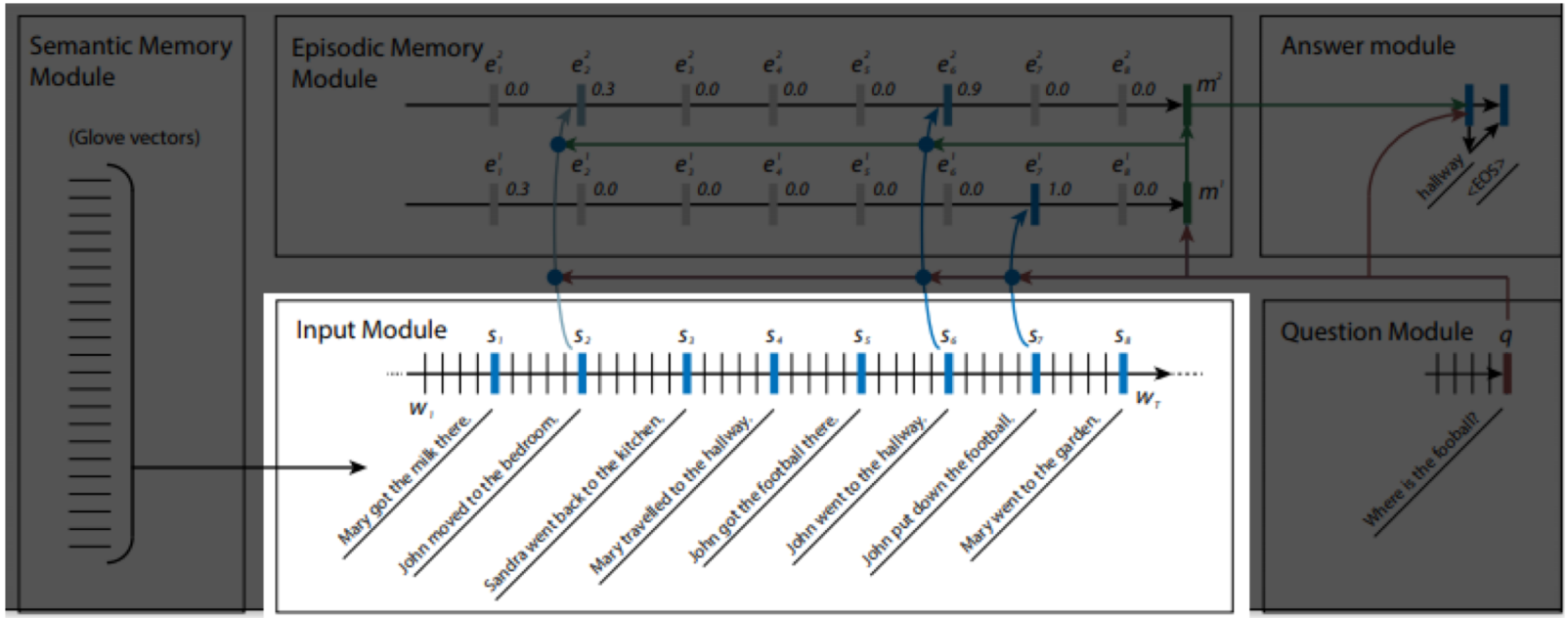
$$\tilde{\boldsymbol{H}}_t = \tanh(\boldsymbol{X}_t \boldsymbol{W}_{xh} + \left(\boldsymbol{R}_t \odot \boldsymbol{H}_{t-1}\right) \boldsymbol{W}_{hh} + \boldsymbol{b}_h)$$

$$\boldsymbol{H}_t = \boldsymbol{Z}_t \odot \boldsymbol{H}_{t-1} + (1 - \boldsymbol{Z}_t) \odot \tilde{\boldsymbol{H}}_t$$



UC Berkeley, STAT 157 L19/6 Gated Recurrent Unit (GRU)

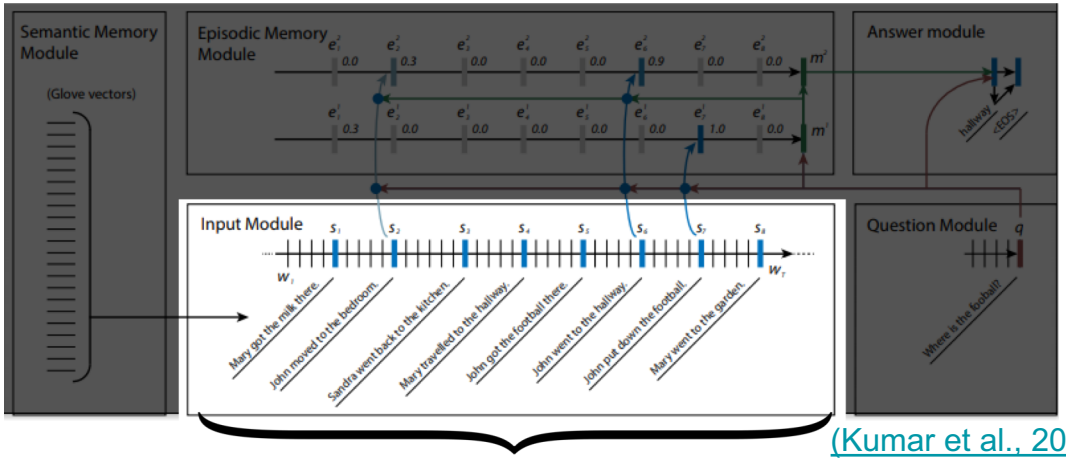
Dynamic Memory Network - **Input Module**



(Kumar et al., 2016)

Input Module

A standard gated recurrent network (GRU) that encodes the input word vectors



(Kumar et al., 2016)

The hidden states at each of the end-of-sentence tokens are the final representations of the input module

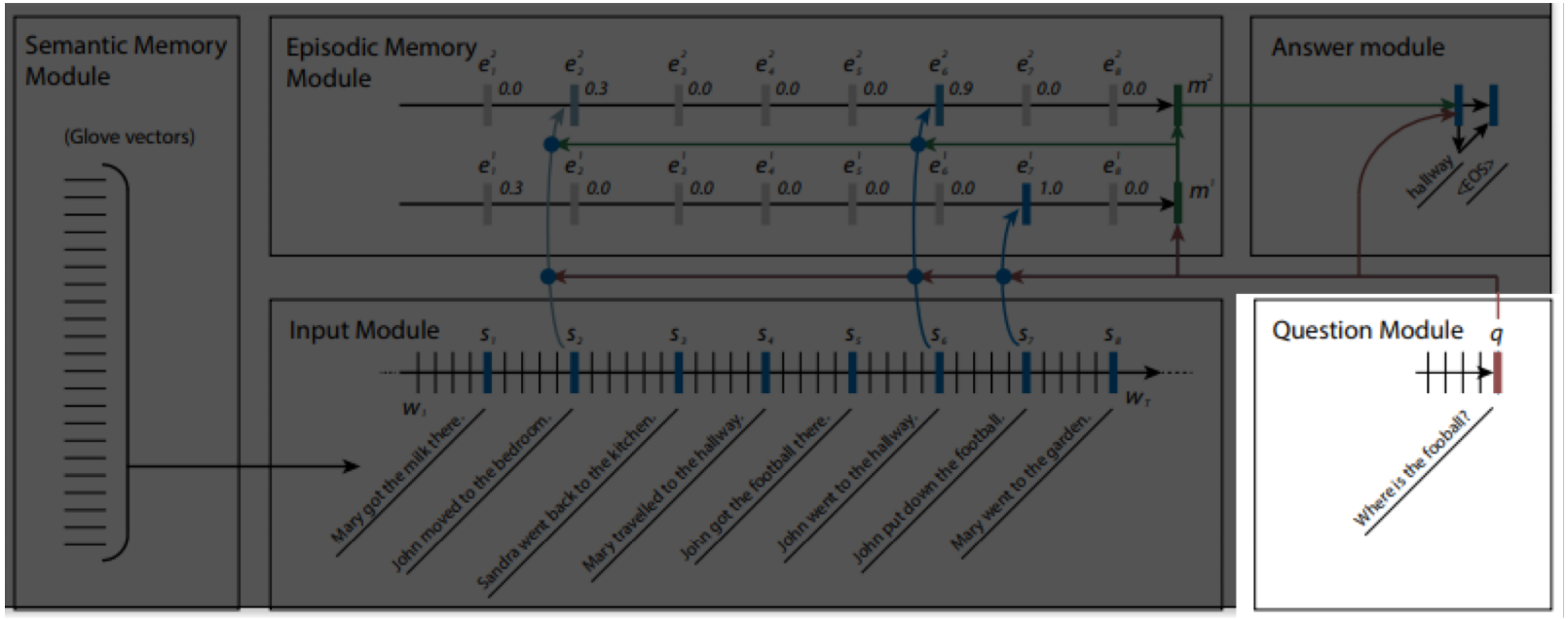
Input: word embeddings (e.g. GloVe or Word2vec)	Concatenate the sentences into a long list of word tokens; insert after each sentence an end-of-sentence token

$$h_t = GRU(x_t, h_{t-1})$$

x_t : input at time t

h_{t-1} : hidden state at time t-1

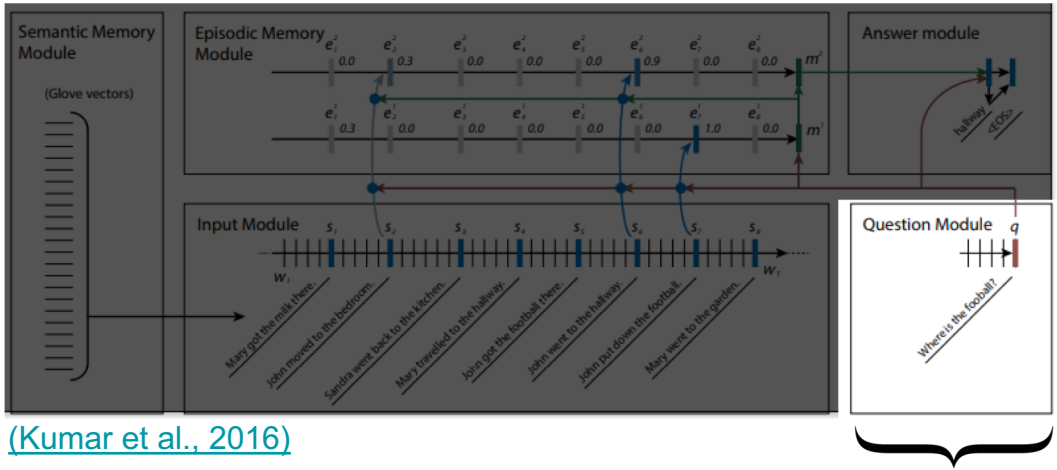
Dynamic Memory Network - **Question Module**



(Kumar et al., 2016)

Question Module

Similar to the input module - a standard GRU that encodes the input question vectors



(Kumar et al., 2016)

Input: word embeddings (e.g. GloVe or Word2vec)

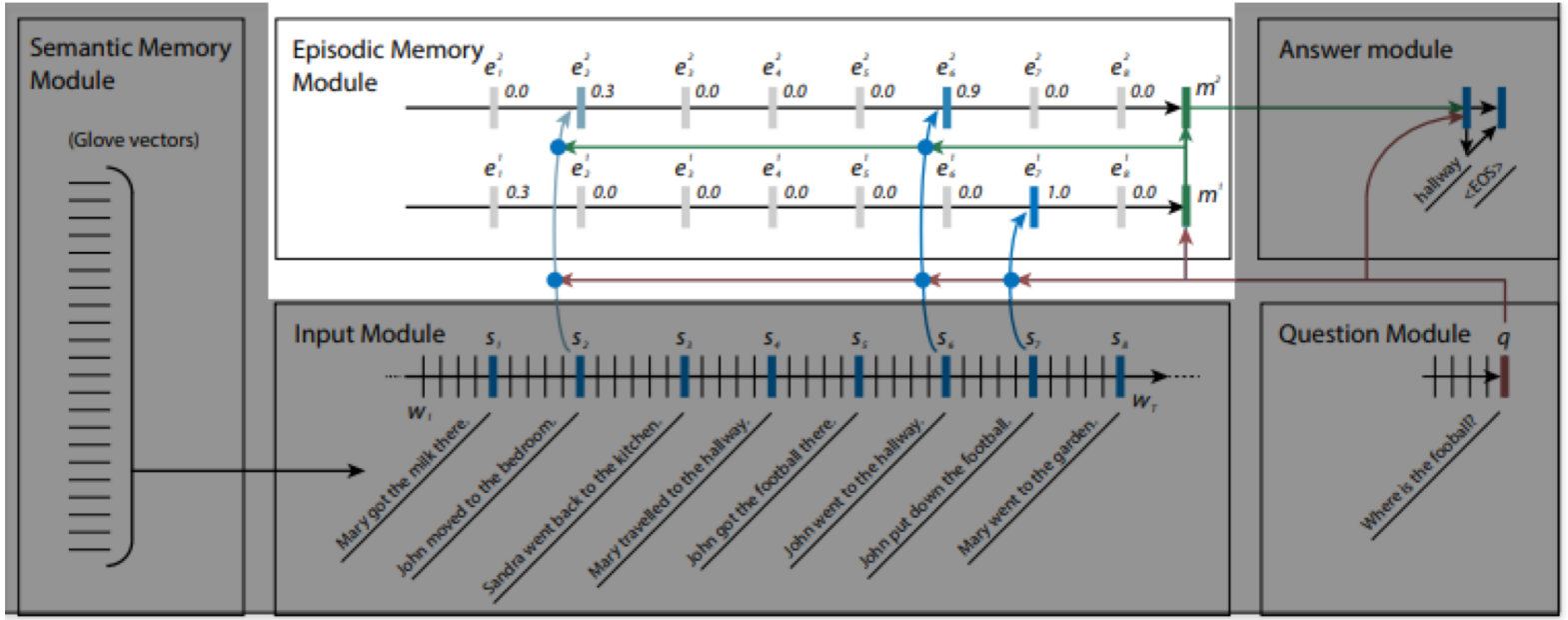
$$q_t = GRU(v_t, q_{t-1})$$

the final hidden state of the recurrent network encoder is then outputted

v_t : word at time t

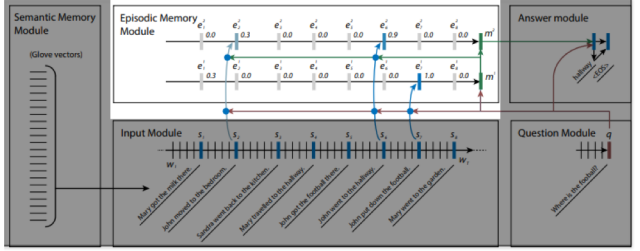
q_{t-1} : hidden state at time t-1

Dynamic Memory Network - **Episodic Memory Module**



(Kumar et al., 2016)

Episodic Memory Module



(Kumar et al., 2016)

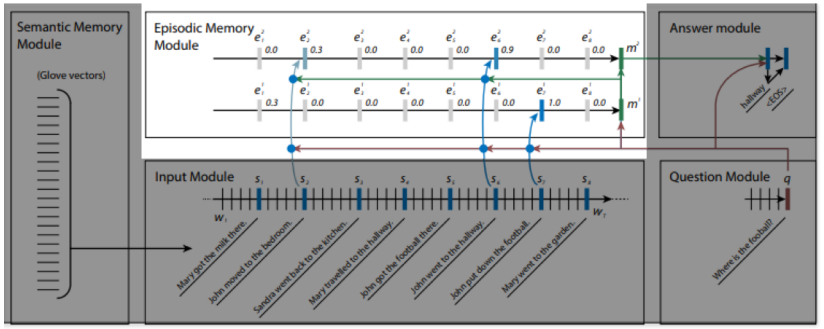
Conducts **multiple iterations** over the input data to update its internal episodic memory

Each iteration, the attention mechanism attends over the sentence embedding / **fact representations** from the input module while taking into account the **question representations** and the **previous memory**.

Here each **fact representations** is assigned a **weight** corresponding to its **relevance to the question** being asked.

A step towards general NLP with Dynamic Memory Networks

Episodic Memory Module



(Kumar et al., 2016)

Different weights may be assigned to the sentence embeddings on different passes. For example,

Input: (1) John is in the playground.
(2) John picked up the football.
(3) Bob went to the kitchen

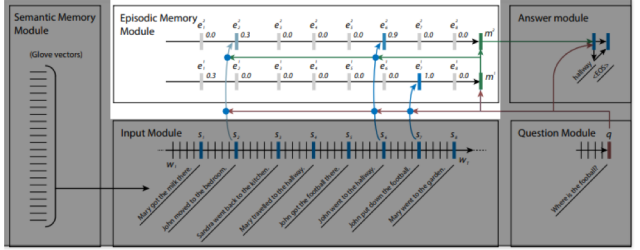
Q: Where is the football?

A: playground

Input sentence (1) is not directly related to the question, it may not be given a high weight on the first pass.

However, on the first pass, the model finds that the football is connected to John, and hence in the second pass sentence (1) is given a higher weight.

Episodic Memory Module



(Kumar et al., 2016)

A step towards general NLP with Dynamic Memory Networks

The first pass uses the **question embedding** q to compute **attention scores** s_o for the sentence embeddings from the input module.

The attention score of sentence s_i is then passed through a **softmax** to obtain g_i , the **weight** given to sentence s_i, and acts as a global gate over the GRU's output at timestep i.

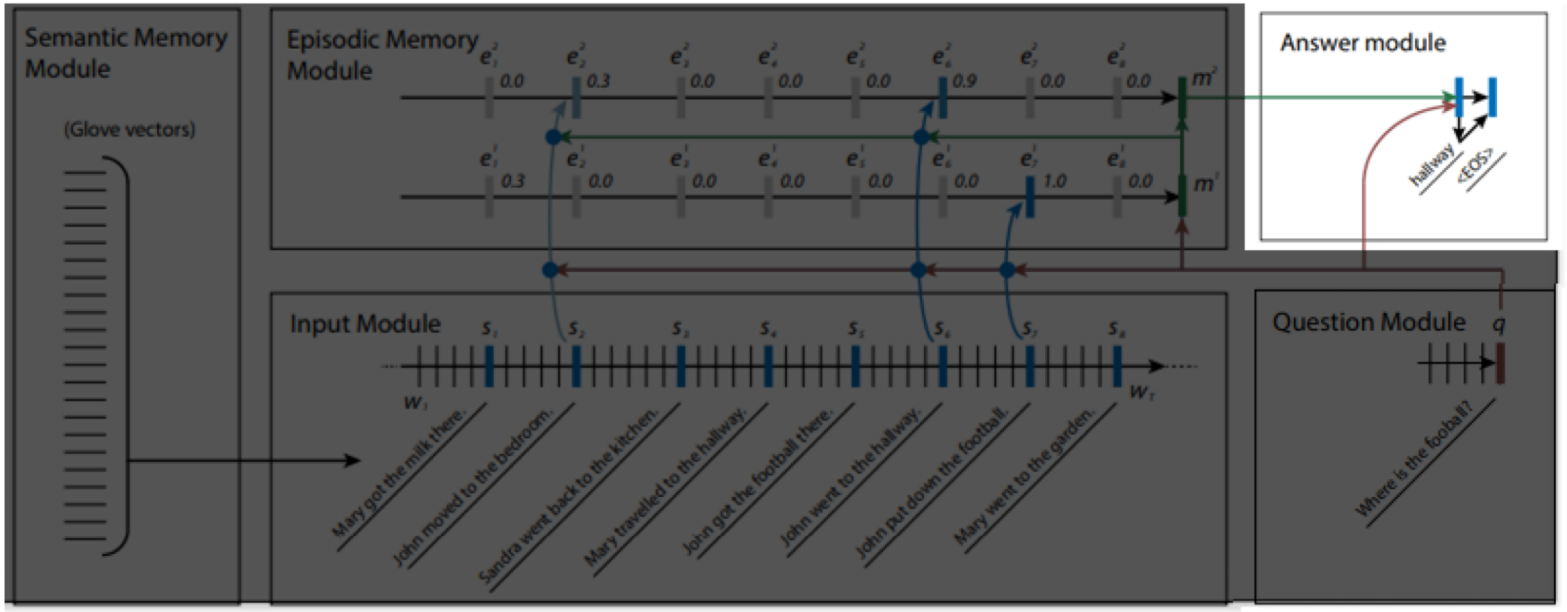
The hidden state for timestep i and episode t is computed as:

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

The last hidden state of the GRU for time t can be viewed as an **agglomeration of the facts** found during time t.

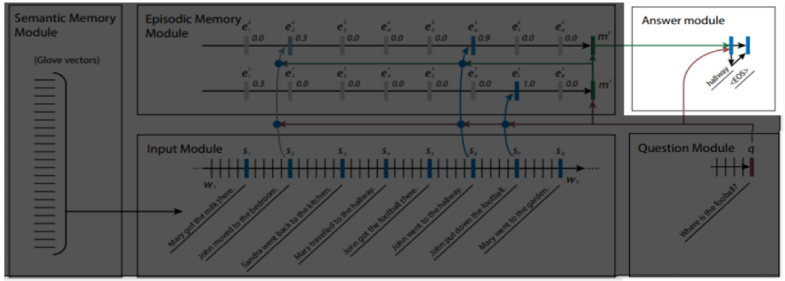
*g_i is calculated using a 2-layer neural network with input related to s_i , q, m_{i-1}, but we will not go into the detail here

Dynamic Memory Network - **Answer Module**



(Kumar et al., 2016)

Answer Module



(Kumar et al., 2016)

Consists of a decoder GRU, whose initial state is initialized to the last memory of the episodic memory module.

At each timestep, the **previously predicted output** y_{t-1} concatenated with the **question embedding** q, the **last hidden state** a_{t-1} are fed as input:

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

A step towards general NLP with Dynamic Memory Networks

Any questions so far?

Outline

- Motivation & Overview
- Dynamic Memory Network [DMN] Explanation
- DMN Experimentations & Results
- Related Work & Comparison
- Conclusion & Discussion

Application to Sentiment Analysis

The DMN achieves state-of-the-art accuracy on the binary classification task, as well as on the fine-grained classification task of the Stanford Sentiment Treebank (SST)

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**

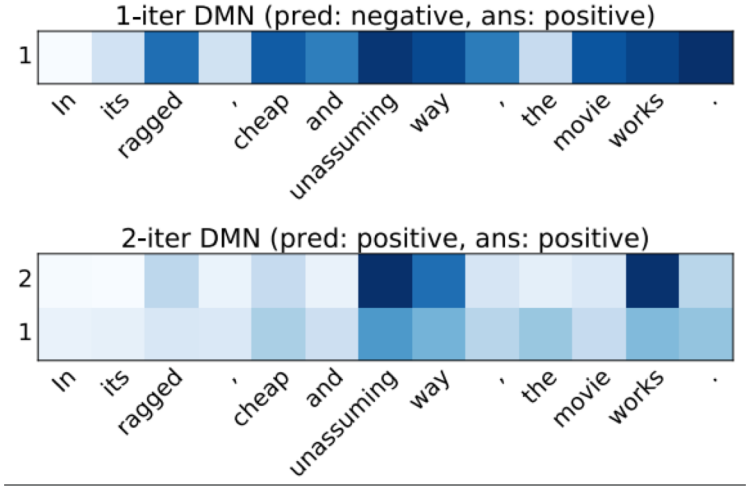
Binary labels	Fine-grained labels
positive, negative	very negative, negative, neutral, positive, or very positive

(Kumar et al., 2016)

Application to Sentiment Analysis - Example

The model pays attention to all the adjectives and produces an incorrect prediction at the first pass.

At the second pass, the model pays significantly higher attention to the positive adjectives on the second pass and produces a correct prediction.



(Kumar et al., 2016)

There Are Also Experiments Done on Other NLP Tasks...

1) Facebook bAbI dataset for Q&A
2) WSJ-PTB (Penn Treebank) for part-of-speech tagging
3) Etc.

We will not go into details about these experimentations.

Let’s Now Look at Some Related Work

- Motivation & Overview

- Dynamic Memory Network [DMN] Explanation
- DMN Experimentations & Results

- Related Work & Comparison
- Conclusion & Discussion

Related Work - Memory Networks (MemNets) (Weston et al. 2015)

Similarities:

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

Differences:

- For input representations, MemNets use bag of word, non-linear 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

Stanford CS224n lecture slides #16

To Conclude…

- Motivation & Overview

- Dynamic Memory Network [DMN] Explanation
- DMN Experimentations & Results

- Related Work & Comparison
- Conclusion & Discussion

Drawing a conclusion!

Conclusion & Discussion

- **Importance**

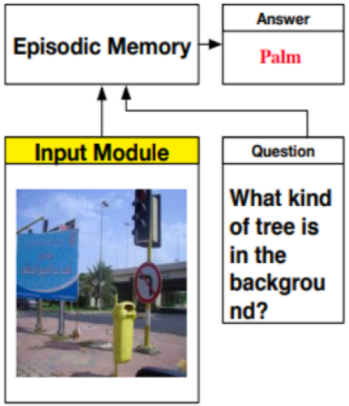
The Dynamic Memory Network (DMN) model is a potentially general architecture for a variety of NLP applications, including classification, question answering, etc.

A single architecture is a first step towards a single joint model for multiple NLP problems.

The DMN is trained end-to-end with one, albeit complex, objective function.

- **Extended work based on this paper**

Post this paper, researchers have tried replacing input module with another which extracted feature vectors from images using a CNN based network. The extracted feature vectors were then fed to the episodic memory module, as before.
(Kumar et al., 2016)



Xiong et al., 2016

Reference

- https://arxiv.org/pdf/1506.07285.pdf
- Natural Language Processing with Deep Learning CS224N/Ling284
- CS224n: Deep Learning for NLP Lecture Notes: Part VIII
- A step towards general NLP with Dynamic Memory Networks
- https://arxiv.org/pdf/1410.3916.pdf