A Quick Summary:
Ask Me Anything: Dynamic Memory Networks for Natural Language Processing

Original Paper: https://arxiv.org/pdf/1506.07285.pdf

22nd Feb 2019

1 Ideas:

In this paper, the authors introduce the Dynamic Memory Network (DMN), a model which consists of an Input, Question, Episodic and Answer module.

(a) Input: Takes in the input knowledge (usually a collection of premises e.g. 1) A dog is a pet. 2) I have a dog)

(b) Question: Self explanatory - just a question to do with the premises. (e.g. Do I have a pet?)

(c) Episodic: A module with attention and memory mechanisms (the main focus of the model) that takes inputs from the Input module and Question module

(d) Answer: A module which takes inputs from the Episodic module and converts it into an answer (a sequence of words, or just a single word).

2 Explanations:

(a) The Input module behaves differently based on how many premises we give it.

i If we give it a single premise with n words, it will come up with n fact representations c, where a fact representation in this case can be thought of to be just a single vector which represents each word.

ii If we give it M premises (which could all be of differing word length), it will come up with M fact representations c.

These fact representations are the output of an RNN:

i Single premise: the fact representations are the output of an RNN at every time step.

ii Multiple premise: the fact representations are the output of the RNN after each premise has ended (Something which was not said earlier is that we feed the multiple premises into the RNN together, as though they were a sequence of multiple sentences).

(b) The Question module only outputs a single vector, which is the final hidden state of the RNN that is located in the Question module.

(c) The Episodic Memory module is a little tricky. Think of it as having an RNN for each episode. Denote episodes using i and timesteps within an episode with t:

i Attention:

$$z(c,m,q) = [c, m, q, c \circ q, c \circ m, |c-q|, |c-m|, c^T W^{(b)} q, c^T W^{(b)} m]$$
$$g = \sigma\Big(W^{(2)} tanh\Big(W^{(1)} z + b^{(1)}\Big) + b^{(2)}\Big)$$

Those equations look intimidating, but z is just a long feature vector which captures the similarities/differences between the fact representations c, previous memory m (take the first memory as the question) and question q. Think of g as a parameter which decides how much of the original hidden state to keep/forget (look at the memory mechanism below).

ii Memory: The memory update is:

$$h_t^i = g_t^i RNN(c_t, h_{t-1}^i) + (1 - g_t^i) h_{t-1}^i$$
$$e^i = h_{T_C}^i$$

There can be a hard limit for the number of episodes which we specify.

(d) The answer module generates an answer given a vector y from the Episodic Memory module. It does this by the following scheme:

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

3 Model:

We look at how the Input module works when it is given multiple premises vs just a single premise.

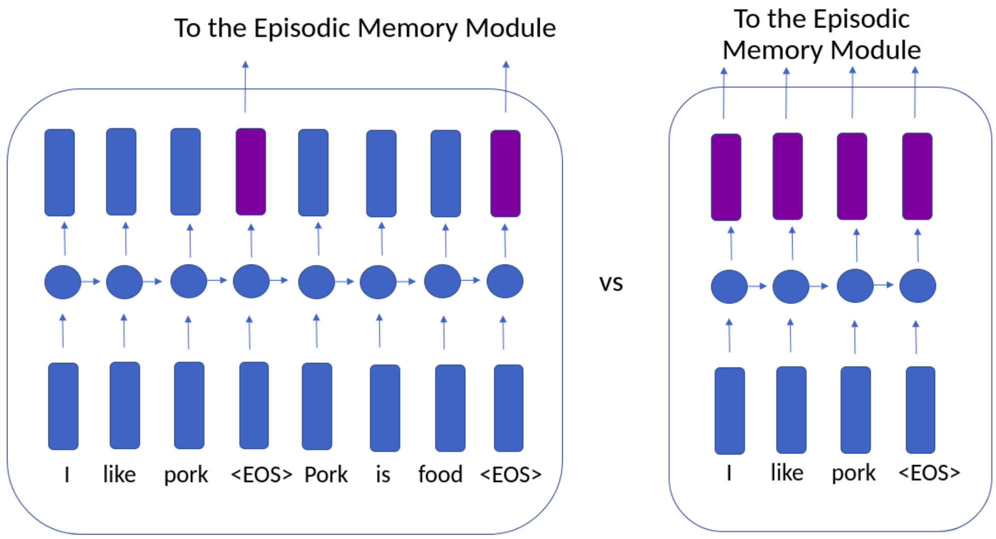

Figure 1: Illustration of how the Input module works when there are multiple premises in the input vs a single premise in the input.

We take a closer look at the Episodic Memory Module during the first episode.

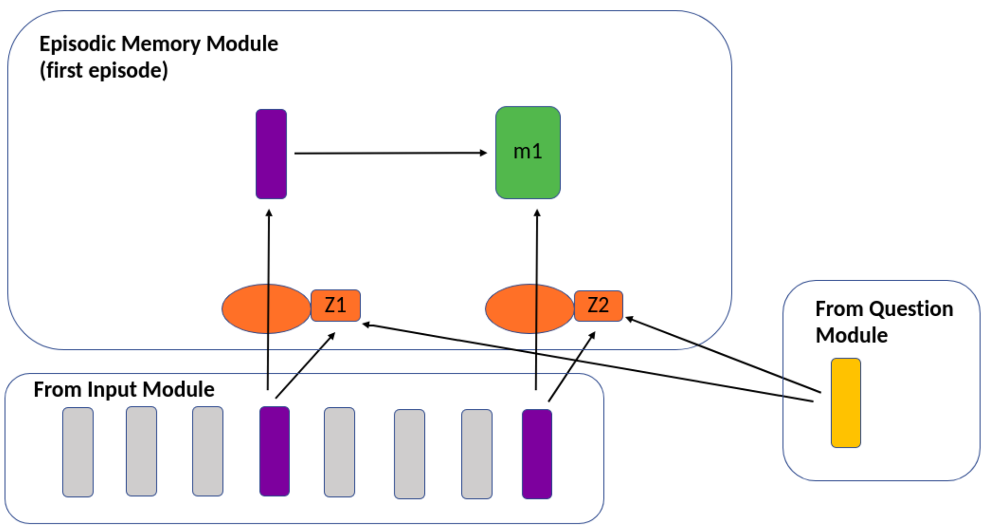

Figure 2: Visualization of the first episode. Orange refers to the attention gates, whilst green refers to the memory once we have gone through the episode. Grey indicates that we pay no attention to these components during this episode. The question module doubles up as the memory m during the first iteration.

Now, we look at the Episodic Memory Module in the second episode to get a concrete understanding of what is going on.

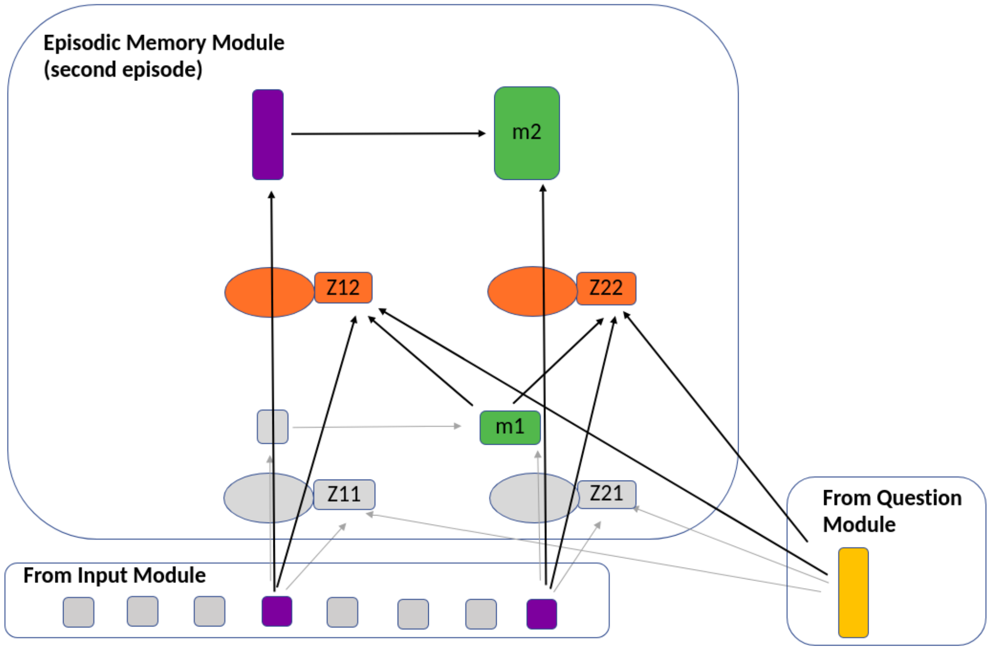

Figure 3: Visualization of the second episode. Orange refers to the attention gates, whilst green refers to the memory once we have gone through the episode. Grey (arrows included) indicates that we pay no attention to these components during this episode.

4 Results:

(a) Slightly better than MemNN when tested on bAbI.

5 Notes:

(a) Introducing models which can be broken down into modules can be helpful during evaluation.

(b) Stanford's NLP course on Youtube has a lecture on this paper.